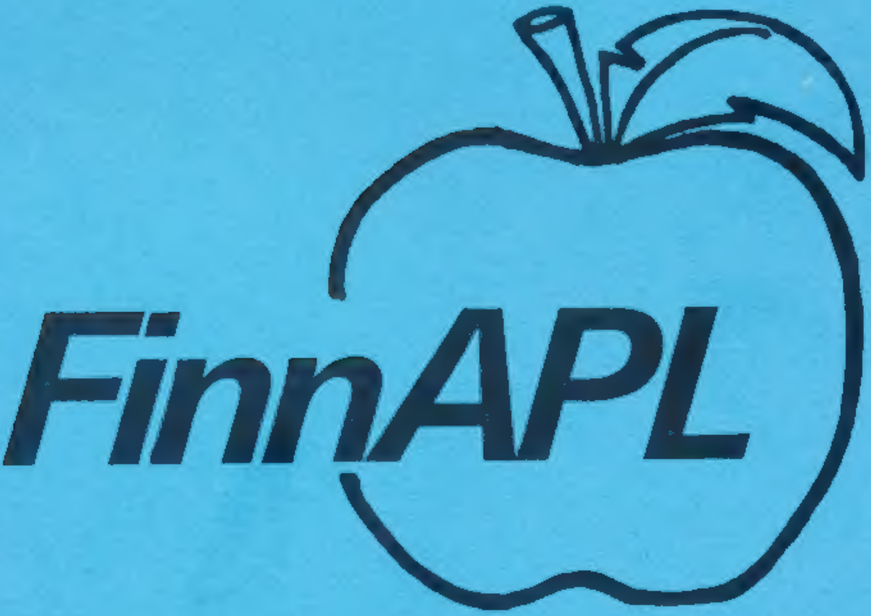
FINNISH APL ASSOCIATION

FINNAPL IDIOM LIBRARY

SECOND EDITION

Jul. 13, 1982



Helsinki 1982

FINNISH APL ASSOCIATION

FINNAPL IDIOM LIBRARY

SECOND EDITION

Jul. 13, 1982

Foreword to First Edition 1

Foreword to Second Edition 2

The Use of Idioms in APL, A. Juvonen 3

How to Use this Idiom Library, T. Laurmaa 9

Table of Functions 13

Idiom List, T. Kunnas (editor) 14

Synonym List 51

Permuted Index 58

Copyright 1981, 1982 Finnish APL Association (FinnAPL)

ISBN 951-99397-9-2

Helsinki 1982

FOREWORD OF FIRST EDITION

The publication of this APL idiom collection was initiated by the FinnAPL Working Group of Programming Techniques in spring 1981, when good ways of teaching and promoting efficient use of APL were being discussed. It is a widely known fact that some idiom containing booklets and APL workspaces exist. However, the working group discovered several deficiencies in the usefulness of these sources: difficult and slow access, the relatively small amount of idioms and the lack of appropriate search methods. These arguments led to the decision of building a versatile, easy-to-use idiom library, from which any APL user can find practical tools for daily work. The goal of this publication is to promote good and efficient APL usage by means of APL idioms, and facilitate and speed up APL programming.

The first and most important source of idioms has been the Working Group itself - the experience and everybody's own idiom reservoirs. The members of the group are:

Arto Juvonen	TMT-Team
Seppo Kaltio	TMT-Team
Timo Kunnas	Helsinki University of Technology
Timo Laurmaa	Bank of Helsinki
Hannu Lehto	Datema
Heimo Penttinen	Finnish Broadcasting Company
Tauno Ylinen	Ministry of Finance

Also the following publications were very useful:

A.J. Perlis, S. Rugaber:
Programming with Idioms in APL,
APL Quote Quad, vol. 9, no. 4, 1979
K.E. Iverson:
Notation as a Tool of Thought
Communications of the ACM, vol. 23, no. 8, August 1980
O.I. Franksen:
Computing - An Economy of Thought
NordDATA81, Copenhagen 1981
R.P. Polivka, S. Pakin:
APL: The Language and Its Usage
Prentice-Hall, 1975
R.A. Smith
A Programming Technique for Non-rectangular Data, APL79 Conference Proceedings, ACM.

The Finnish APL Association is pleased to receive any comments and new idioms in order to increase the versatility and quality of this idiom library.

FOREWORD OF SECOND EDITION

The second edition of the FinnAPL Idiom Library differs from the first one in several aspects: the number of idioms has been increased from 452 to 631, most of the idioms have been tested with a computer, many errors have been corrected, some explanations have been made easier to understand, and a function-based index has been included. The publication is also introduced in a more compact version (FinnAPL Pocket Idiom Library).

The working group for this edition has been:

Akif Ali	Helsinki University of Technology
Timo Kunnas	Helsinki University of Technology
Timo Laurmaa	Bank of Helsinki

FinnAPL wishes to thank for the numerous contributions by APL users from several countries, and is still pleased to receive comments and new idioms to help in developing future editions of the library.

Finnish APL Association FinnAPL
P.O. Box 1005
SF-00101 Helsinki 10
FINLAND

THE USE OF IDIOMS IN APL

Arto Juvonen
TMT-Team

1. INTRODUCTION

Idiom is a speech form that is peculiar within the usage of a given language. In APL, an idiom is a meaningful primitive structure constructed from primitive functions. Idiom is characterized by ease of recognition, frequency of occurrence and flexibility in use. Due to APL's conciseness and functional nature the language contains a large number of useful idioms, whose familiarity makes the programming effort more efficient and unravels programs made by others.

An idiom is an APL primitive construct like the APL primitive and derived functions because of its presence in the language. However, an idiom has a special connotation, that cannot be directly derived from its constituents. Once the idiom is learned, the meaning can be perceived from the program line without analyzing the structure function by function. Let us take as an example the 'if'-idiom ⌈⍳ in a conditional branch statement. Due to the expression →LINE⌈⍳COND the program branches to line LINE, if COND (1 or 0) is true. Otherwise execution continues at the next line. The expression might be difficult to decipher, if the idiom is not familiar. After learning the behavior of the idiom the use and understanding of it is trivial.

One of the main characteristics of an idiom is its conciseness. It is easy to remember and recognize and quick to type. Because of this it is not necessary to define an idiom as a function, although this is quite common. Idioms are often more useful as part of another expression. By writing the idiom straight to the function line the naming problem can be avoided. A person acquainted with the idiom does not have to know in which workspace and with which name the expression is saved, and does not have to copy the function into the workspace. Also the problem of having a different object of the same name in the workspace can be avoided. Moreover, the reader of the program does not have to recognize the name. It is also useful to use idiomatic expressions in the library subroutines instead of the equivalent

defined functions to avoid chained calls and thus group definitions or copying of defined library functions one by one during the first application programming test run.

The difference between an idiom and a defined function can easily be seen. An important difference is their length. Idioms are almost always expressed naturally in one line. A defined function may contain more complicated, e.g. iterative structures. If an idiom is embedded in a defined function, it is useful to write it in a general form to allow wider use. Thus in a defined function the idiom is hidden in the jungle of shape and reshape functions that guarantee the compatibility of the arguments. If the idiom is written as part of the function line, the simplest form according to the current arguments can be used. There are also idioms that cannot be written as a function because of the great number of arguments or some other special feature. Such idioms are e.g. templates, that can be used to create new idioms.

2. THE USE OF IDIOMS IN WRITING AND READING PROGRAMS

No language contains a perfect set of functions for every need encountered in solving problems. APL idioms can often be used as APL primitive functions. They make the language more expressive and bring the language nearer to the human thought structures. Idioms act as an intermediate language in man-machine communication. Thus, mastering a group of basic idioms is necessary for efficient APL programming allowing the lines to be written from left to right and not a function at a time from right to left.

Idioms act also as a means of convaying thougths between the program writer and the reader. Understanding a program is a discontinuous process. The meaning of a line is first unclear. After initial research the reader can make a hypothesis of the meaning. Additional studying verifies the model. Insight is more easily attained if the program writer and reader share a large number of idioms. Not knowing the idiom written by someone else interrupts understanding of the program flow. The reader can study the result of functions applied to sample arrays element by element, but making a hypothesis of the meaning of the expression may be very difficult.

Thus both in writing and reading programs

mastering a large set of idioms is an invaluable tool. Idioms should be learned early while learning other language fundamentals, as this helps to organize the students understanding of the numerous possibilities of the primitive functions and their combinations. Idioms are best learned by encountering them in different environments and by using them. New idioms can be found by using one-line solutions, as this helps avoiding splitting of data and thought.

3. EXAMPLES OF IDIOMS

Idioms can be classified for example in the following manner:

1. Structural
2. Sorting
3. Searching
4. Tests
5. Forming boolean vectors
6. Text processing
7. Numerical
8. Object generators
9. Subvector processing
10. Templates

Changing the structure of an array is common because of the compatibility requirements of functions. A typical example is forming a vector into a one line matrix V←(1,⍴V)⍴V or a 'column vector' V←⍉(1,⍴V)⍴V. A numeric vector can be reshaped also with the idiom V←V∘.+,0. Often operating with higher dimensional arrays may be cumbersome. E.g. multiplying an array A of shape k,m,n with array B of shape k,m,l,n so that the elements along the second axis of A are also used for multiplying the elements along the third axis of B necessitates forming a new axis in A after the second axis. The new axis is first shaped to contain a dimension of length one. The dimension is replicated l times by indexing:
S←(1,⍴A)[2 3 1 4]⍴A and S←S[;;(⍴B)[3]⍴1;]. An array is restructured often by the transpose function T⍉A. The element T[I] specifies, where the i:th axis of the argument array will be located in the resulting array. Thus forming a new axis after the i:th axis and replicating it n times can be generally stated with the idiom (⍋⍋(I+1),⍳⍴⍴A)⍉(N,⍴A)⍴A.

Sorting arrays is based on the grade primitive function ⍋. The task of idioms is to create suitable numeric arguments for the grade function.

For example a word list can be sorted with the idiom I←⍋(⍴A)⊥⍉A⍳M, where A contains the alphabet used and M the words to be sorted. By using this idiom a sorting of 8 to 10 columns can be performed at a time. Sorting a wider matrix or taking other factors into consideration requires sorting to be started with the least significant sorting factor and indexing the next factor with the resulting grade vector. E.g. forming the grade vector I by using the first two columns of a numeric matrix M would be performed by using first the second column I←⍋M[;2] and then the first one I←I[⍋M[I;1]].

Often we have to sort numeric codes in vector A to the corresponding sequence S e.g. for summing. The required grade vector can be formed with the idiom I←⍋S⍳A. A[I] can be used to get the lengths of the subvectors to be summed.

Searching is often based on location function (⍳) or compression of indices L/⍳⍴L. The location function can be used e.g. to locate the index of the first positive element in vector V: (V>0)⍳1 or the index of the first occurrence of vector A in matrix M: (M∧.=A)⍳1.

Test idioms are very frequent. Good examples are the test of numericity of an array A: 0∊0\0⍴A, and the test of emptiness: 0∊⍴A, and the test of equalness of all elements in vector A: ∧/A=1↑A.

Forming a modification of a logical vector is often fundamental when using the APL way of solving problems. E.g. the following transformations can be performed to a logical vector (array):

```
Turn on all 0's after the first 1
L←                        0 0 1 0 1 0 1
∨\L                       0 0 1 1 1 1 1

Leave on the 1's preceding the first 0
L←                        1 1 1 0 1 0 1
∧\L                       1 1 1 0 0 0 0

Leave on only the first 1
L←                        0 0 1 0 1 1 1
<\L                       0 0 1 0 0 0 0

Create a vector of running even parity
L←                        1 0 0 1 1 0 1
≠\L                       1 1 1 0 1 1 0
```

```
Leave on the first 1 on each group of 1's
L←                          0 1 1 0 1 1 1
L>¯1↓0,L                    0 1 0 0 1 0 0
Generally L>((-⍴⍴L)↑¯1)↓0,L

Leave on the first group of 1's
L←                          0 1 1 0 1 1 0
L∧∧\L=∨\L                   0 1 1 0 0 0 0

Leave on the groups  of 1 in B  indicated  by
the corresponding 1 in vector L
L←                          0 1 0 0 0 0 1 0 1
B←                          1 1 1 0 1 1 0 1 1
B∧A∊(L∧B)/A←+\B>¯1↓0,B      1 1 1 0 0 0 0 1 1
```

Text processing often aims at finding a character or a character string satisfying given conditions. Because APL primitive functions use a character as an element, the text and word processing area contains many idioms to perform higher level operations. Some examples are deletion of blanks from a character vector V: `(V≠' ')/V`, deleting leading blanks `(∨\V≠' ')/V` and trailing blanks: `(¯1↑(V≠' ')/⍳⍴V)/V`. Replacing each element A in vector V with B: `V[(V=A)/⍳⍴V]←B` is simple, and the idiom `V[L/⍳⍴L]←B[(L←I≤⍴A)/I←A⍳V]` can be used, if A and B are vectors. The occurrences of vector A in vector V can be located with the idiom
`I←(V[I∘.+¯1+⍳⍴A]∧.=A)/I←(I=1↑A)/⍳⍴I←(1-⍴A)↓V`.
In it the possible starting locations are found by the initial letter and the length of A. The substrings of V starting from these indices are compared with A.

Vectors are often the most convenient way to handle data because of flexible selection and indexing. A vector is often divided into variable length parts (subvectors) that must be operated on analogously. In the following examples a boolean vector P indicates the partitioning of the argument vector.

```
P←1 0 0 0  1 0  1
A←1 2 3 4  5 6  7  and  L←1 1 0 1  0 0  1

The lengths of the parts
Z←Z-¯1↓0,Z←(1⌽P)/⍳⍴P      4 2 1

The sum of the elements of parts in A
Z←Z-¯1↓0,Z←(1⌽P)/+\A      10 11 7

The or-reduction of the parts in vector L
Z←(P/L)≥Z/1⌽Z←(P∨L)/P     1 0 1
```

A template is a model that can be used to create useful structures. In the following templates `⍺` stands for a suitable function or another idiom. Template `(⍴A)⍴⍺,A` means raveling array A, operating on it and reshaping it back to the original shape. By using this template and a suitable idiom we can e.g. change the elements A in array S to B: `V[(V=A)/⍳⍴V←,S]←B` and `S←(⍴S)⍴V`.

Using numerical calculation instead of conditional branching is often useful. E.g. idiom `(X×B)+Y×~B` can be used to select an element from X if the corresponding element in B is true, otherwise from Y. The Idiom `X*1 ¯1[B+1]` can be used to negate the elements of X indicated by B.

The template `A⍉B∘.⍺C` encourages to use outer product to compute all needed combinations and to select the needed ones with the diagonal transpose... For example adding the vector X to the columns of matrix M can be performed with the idiom `1 1 2⍉X∘.+M`.

HOW TO USE THIS IDIOM LIBRARY

Timo Laurmaa
Bank of Helsinki

1. THE PARTS OF THE IDIOM LIBRARY

1.1 Idiom list

The library is based on an idiom list containing 631 APL idioms. Every idiom consists of an APL expression, a brief verbal description, the description of arguments and parameters, and an index number. The list is sorted in a 'functionally alphabetical' order; therefore two very similar idioms will not be far from each other. The functionally alphabetical order is based on the order of functions listed in the Table of Functions so that e.g. expressions containing ⍋ are found in the beginning of the list and so on.

The argument description states the requirements set to the variables in the expression (arguments or parameters). The most common requirements include information of the rank or shape of the array as well as the type, according to the following rules:

- Naming convention of data objects:
  -- X,Y arguments
  -- G,H parameters
  -- A,B intermediate results etc.
- Data type expressions (in argument descriptions):
  -- C = character
  -- B = logical (boolean)
  -- I = integer
  -- D = any numeric data type
  -- A = any data type, no limitations
- Representation of the ranks of arrays:
  -- 0 = scalar
  -- 1 = vector
  2 = matrix
  -- = (empty) no limitations; the shape is mentioned separately, if necessary
- Examples
  -- X←D2 argument X is a numeric matrix
  -- G←I0 parameter G is an integer scalar

1.2 Permuted index

The bulk of this idiom library consists of an alphabetical index sorted according to a so-called permuted index (keyword in context) method. This method is used in order to facilitate finding appropriate idioms with the assistance of typical keywords. A verbal description with n words will usually occur in n places in the index, at the place of each word of the description. This provides an easy way to find a suitable idiom even if its description is not completely known. The index contains a reference to the index numbers in the idiom list. If a description corresponds to two idioms with numbers clearly unequal, it is likely that the desired result can be achieved with two highly different solutions (nearly equal numbers would indicate similarly built expressions).

1.3 Synonym list

Same results can be achieved in several different ways in APL. This list is a collection of certain idioms from the idiom list that perform (at least approximately) similar operations. (They may differ from each other for instance by requirements set for their arguments.) The synonym list gives a good insight of how to produce similar results in many different ways in APL. Studying these differences will certainly help widening the understanding of APL idioms and expressions.

2. LIMITATIONS OF THE IDIOM LIBRARY

APL idiom library has been designed for the most common programming circumstances. Consequently, some idioms will not work as desired in every special situations. The idioms will generally work with at least the default values of a VS APL empty workspace, which usually apply also in application workspaces. Possible circumstances for unexpected behaviour of certain idioms might be for instance

- ⎕PP←4 Printing precision is small
- ⎕CT←0.1 Comparison tolerance in high

The index origin ⎕IO deserves special consideration. As the default value is one, every idiom (unless the desription states differently) will work if ⎕IO = 1. Several idioms include ⎕IO in the APL expression, which means that the idiom works with both possible ⎕IO values. Exception

cases (works only if ⎕IO = 0) are reported separately in the verbal description. The great majority of the idioms will work in the expected way, independent of parameter values.

This publication intends to present only efficient idioms that promote good APL programming style. It is, however, sometimes difficult to decide if an APL expression will be efficient in every situation. Therefore, especially in handling very large data objects there is a danger that huge amounts of cpu time will be spent or that a WS FULL occurs unexpectedly. If the idiom is good in principle ((X⍳X)=⍳⍴X is a typical example), but in some conditions may turn out to be inefficient, this is mentioned in the verbal description. Avoiding WS FULL problem is possible for instance with the thumb rule stating that the outer product ∘. may easily cause a WS FULL if large data objects are being handled.

Some idioms include numeric or character constants which make the idioms more recognizable but seem to reduce their generality. For instance, the idiom DIVIDING A 400×12 CHARACTER MATRIX INTO ONE PAGE contains the constants 400 and 12, which clearly have to be modified according to the column width and page length.

Certain idioms contain the delta function ∆ needed when chaining expressions. If these idioms are used, the workspace should contain this function defined in the following way:

```
      ∇Z←A ∆ B
  [1]   Z←A
      ∇
```

This function causes the rightmost expression to be processed first. The leftmost expression will then be executed independently of the result of the previous one.

3. EXAMPLES OF THE USE OF THE IDIOM LIBRARY

3.1 Finding an idiom to perform a desired task

If we want to use the idiom library to find an expression that underlines a text vector, we can examine words UNDERLINE, TEXT and VECTOR in the index. If the desired idiom belongs to the library, some of these key words will very likely be found in the index. As related idioms are described as UNDERLINING A STRING and UNDERLINING

WORDS, key word UNDERLINING will show where a suitable idiom can be found.

3.2 How can decode ⊤ be used

Because the idiom list is sorted according to a functional order, most idioms using ⊤ are grouped in the same part of the list, near its beginning. Studying idioms in this way will help learning new ways of applying APL functions and thus widen the APL way of thinking.

3.3 Removing duplicate lines or elements

The index part contains several idioms under the keyword DUPLICATE. These idioms have been collected to the synonym list. By comparing these idioms it is possible to find an expression that will work properly and efficiently in the desired situation, like finding different words from a word matrix.

3.4 What does expression A∘.+,0 do

Idioms based on the outer product can easily be found in the idiom list. The verbal description of idiom X∘.+,0 shows that this is a way to form a one-column matrix from a vector.

TABLE OF FUNCTIONS

1 ... 37 GRADE UP ⍋
38 ... 44 GRADE DOWN ⍒
45 ... 63 ENCODE ⊤
64 ... 84 DECODE ⊥
85 ... 101 EXECUTE ⍎
102 ... 111 FORMAT ⍕
112 ... 120 BRANCH →
121 ... 123 ROLL / DEAL ?
124 ... 128 MATRIX INVERSION / MATRIX DIVISION ⌹
129 ... 134 GEOMETRICAL ○
135 ... 142 FACTORIAL / BINOMIAL !
143 ... 204 OUTER PRODUCT ∘.ω
205 ... 246 INNER PRODUCT α.ω
247 ... 310 SCAN ω\ ω⍀
311 ... 374 REDUCTION ω/ ω⌿
375 ... 412 TAKE ↑
413 ... 448 DROP ↓
449 ... 449 CEILING / MAXIMUM ⌈
450 ... 462 FLOOR / MINIMUM ⌊
463 ... 479 MAGNITUDE / RESIDUE |
480 ... 490 EXPAND \
491 ... 513 COMPRESS /
514 ... 527 TRANSPOSE ⍉
528 ... 537 REVERSE / ROTATE ⌽ ⊖
538 ... 548 MEMBER OF ∊
549 ... 563 INDEX GENERATOR / INDEX OF ⍳
564 ... 569 LOGICAL ~ ∨ ∧ ⍱ ⍲
570 ... 575 COMPARISON < ≤ = ≥ > ≠
576 ... 598 RAVEL / CATENATE ,
599 ... 609 AXIS / INDEXING [
610 ... 618 SHAPE / RESHAPE ⍴
619 ... 628 ARITHMETIC + - × ÷
629 ... 631 MISCELLANEOUS

IDIOM LIST

```
---
    GRADE UP  ⍋

 1  ⌊.5×(⍋⍋X)+⌽⍋⍋⌽X                                          X←D1
    Ascending cardinal numbers (ranking, shareable)

 2  Y[A⍳⌈\A←⍋A[⍋(+\X)[A←⍋Y]]]                            X←B1; Y←D1
    Maximum scan (⌈\) over subvectors of Y indicated by X

 3  Y[A⍳⌈\A←⍋A[⍋(+\X)[A←⍒Y]]]                            X←B1; Y←D1
    Minimum scan (⌊\) over subvectors of Y indicated by X

 4  Y[⍋Y]∧.=X[⍋X]                                        X←D1; Y←D1
    Test if X and Y are permutations of each other

 5  X[A[⍋(+\(⍳ρX)∊+\⎕IO,Y)[A←⍋X]]]          X←D1; Y←I1; (ρX) ↔ +/Y
    Sorting subvectors of lengths Y

 6  Y[A[X/⍋(+\X)[A←⍋Y]]]                                 X←B1; Y←D1
    Minima (⌊/) of elements of subvectors of Y indicated by X

 7  A[⍋(+\X)[A←⍋Y]]                                      X←B1; Y←D1
    Grading up subvectors of Y indicated by X

 8  (ρX)ρ(,X)[⎕IO+A[⍋⌊A÷¯1↑ρX]]  Δ  A←(⍋,X)-⎕IO                X←D2
    Sorting rows of matrix X into ascending order

 9  (ρX)ρ(,X)[A[⍋(,⍉(⌽ρX)ρ⍳1↑ρX)[A←⍋,X]]]                      X←D2
    Sorting rows of matrix X into ascending order

10  (⍋⍋(G+1),⍳ρρX)⍉(Y,ρX)ρX                       G←I0; Y←I0; X←A
    Adding a new dimension after dimension G Y-fold

11  (Y,X,Z,...)[⍋⍋G]                X←A1; Y←A1; Z←A1; ... ; G←I1
    Merging X, Y, Z ... under control of G (mesh)

12  (X,Y)[⍋⍋G]                                X←A1; Y←A1; G←B1
    Merging X and Y under control of G (mesh)

13  ⍋⍋X                                                      X←D1
    Ascending cardinal numbers (ranking, all different)

14  Y[A[X/⍋(+\X)[A←⍒Y]]]                                 X←B1; Y←D1
    Maxima (⌈/) of elements of subvectors of Y indicated by X

15  A[⍋(+\X)[A←⍒Y]]                                      X←B1; Y←D1
    Grading down subvectors of Y indicated by X

16  (Y,X)[⍋⍒G]                                X←A1; Y←A1; G←B1
    Merging X and Y under control of G (mesh)

17  ⍋⍒X                                                      X←D1
    Descending cardinal numbers (ranking, all different)
```

```
18  X[⍋(1+⍴Y)⊥Y⍳⍉X;]                               X←A2; Y←A1
    Sorting rows of X according to key Y (alphabetizing)

19  X[⍋+⌿A<.-⍉A←X,0;]                                     X←D2
    Sorting a matrix into lexicographic order

20  X[⍋X]∧.=⍳⍴X                                           X←I1
    Test if X is a permutation vector

21  Y[⍋X++\X]                                       X←B1; Y←A1
    Rotate first elements (1⌽) of subvectors of Y indicated by X

22  1↑⍋X                                                  X←D1
    Index of (first) minimum element of X

23  ¯1↑⍋X                                                 X←D1
    Index of (first) maximum element of X

24  X[(⍋X)[⌈.5×⍴X]]                                       X←D1
    Median

25  (X,'''')[(⎕IO+⍴X)⌊⍋(⍳⍴X),(''''=X)/⍳⍴X]                 X←C1
    Doubling quotes (for execution)

26  (X,'*')[(⎕IO+⍴X)⌊⍋(⍳⍴X),(Y×⍴G)⍴G]           X←C1; Y←I0; G←I1
    Inserting Y *'s  into vector X after indices G

27  (⍴X)≥⍋(⍳⍴X),Y                                    X←D1; Y←I1
    Expansion vector with zero after indices Y

28  (X,A⍴H)[⍋(⍳⍴X),A⍴Y] Δ A←G×⍴,Y         X←A1; Y←I1; G←I0; H←A0
    Catenating G elements H after indices Y in vector X

29  ((A⍴H),X)[⍋(A⍴Y),⍳⍴X] Δ A←G×⍴,Y       X←A1; Y←I1; G←I0; H←A0
    Catenating G elements H before indices Y in vector X

30  ⍋Y⍳X                                              Y←A1; X←A1
    Grade up according to key Y

31  A Δ A[⍋G]←A←Y,X                             X←A1; Y←A1; G←B1
    Merging X and Y under control of G (mesh)

32  X[⍋Y[X]]                                          X←I1; Y←D1
    Sorting indices X according to data Y

33  X[⍋X[;Y];]                                            X←D2
    Sorting a matrix according to Y:th column

34  ⍋X×¯1 1[Y]                                        X←D1; Y←I0
    Choosing sorting direction during execution

35  X[⍋X]                                                 X←D1
    Sorting X into ascending order
```

36 `Y[⍋X]` X←A1; Y←A1
Sorting Y according to X

37 `⍋X` X←I1
Inverting a permutation

GRADE DOWN ⍒

38 `X[⍒+⌿A<.-⍉A←X,0;]` X←D2
Sorting a matrix into reverse lexicographic order

39 `X[⌽⍒+\(⍳⍴X)∊+\⎕IO,Y]` X←D1; Y←I1
Reversal (⌽) of subvectors having lengths Y

40 `Y[⌽⍒+\X]` X←B1; Y←A1
Reversal (⌽) of subvectors starting at indices X

41 `(+/X)↑⍒X` X←B1
Indices of ones in logical vector X

42 `X[⍒' '≠X]` X←C1
Moving all blanks to end of text

43 `X[⍒Y]` X←A1; Y←B1
Moving elements satisfying condition Y to the start of X

44 `X[⍒X]` X←D1
Sorting X into descending order

ENCODE ⊤

45 `⍕10⊥((1+⌈2⍟⌈/,X)⍴2)⊤X` X←I0
Binary format of decimal number X

46 `1 0⍕10 10⊤1-⎕IO-⍳X` X←I0
Helps locating column positions 1..X

47 `⌽((0,⍳⍴X)∘.=+⌿~A)+.×(-X)×.*A←((⍴X)⍴2)⊤¯1+⍳2*⍴X` X←D1
Polynomial with roots X

48 `⎕IO+(⍴X)⊤-⎕IO-(,(X=(⍴X)⍴⌈⌿X)∧X=⍉(⌽⍴X)⍴⌊/X)/⍳×/⍴X` X←D2
Index pairs of saddle points

49 `⍉'0123456789ABCDEF'[⎕IO+((⌈⌈/16⍟,X)⍴16)⊤X]` X←I
Conversion from decimal to hexadecimal

50 `(,X)/1+A⊤¯1+⍳×/A←⍴X` X←C2
Transforming connectivity matrix X into a connectivity list

51 `⎕IO+(⍴X)⊤(⍳×/⍴X)-⎕IO` X←A
Matrix of all indices of X

```
52  ((⌈2⍟1+X)⍴2)⊤0,⍳X                                        X←I0
```
All binary representations up to X (truth table)

```
53  ((1+⌊10⍟X)⍴10)⊤X                                         X←I0
```
Digits of X separately

```
54  ((1+⌊Y⍟X)⍴Y)⊤X                                   X←D0; Y←D0
```
Representation of X in base Y

```
55  ⎕IO+(⍴X)⊤(-⎕IO)+(,X∊Y)/⍳⍴,X                        X←A; Y←A
```
Indices of elements Y in array X

```
56  ,' ',⍉'0123456789ABCDEF'[⎕IO+16 16⊤-⎕IO-⎕AV⍳X]          X←C1
```
Conversion of characters to hexadecimal representation (⎕AV)

```
57  ⍉(3⍴100)⊤X                                                X←D
```
Separating a date YYMMDD to YY, MM, DD

```
58  ⎕IO+(X,Y)⊤(⍳X×Y)-⎕IO                             X←I0; Y←I0
```
All pairs of elements of ⍳X and ⍳Y

```
59  ((⍴X)⍴2)⊤¯1+⍳2*⍴X                                        X←A1
```
Matrix for choosing all subsets of X (truth table)

```
60  (X⍴2)⊤¯1+⍳2*X                                            X←I0
```
All binary representations with X bits (truth table)

```
61  1+Y⊤X                                             X←D; Y←D0
```
Incrementing cyclic counter X with upper limit Y

```
62  0 1⊤X                                                     X←D
```
Integer and fractional parts of positive numbers

```
63  10 100 1000⊤X                                             X←I
```
Decoding numeric code ABBCCC into a matrix

DECODE ⊥

```
64  A Δ A[3 6]←':' Δ A←⍕1000⊥3↑3↓⎕TS
```
Representation of current time (24 hour clock)

```
65  A Δ A[5 8]←'-' Δ A←⍕1000⊥3↑⎕TS
```
Representation of current date (descending format)

```
66  →Y[1+2⊥X]                                        Y←I1; X←B1
```
Case structure with an encoded branch destination

```
67  G⊥Y⌹X∘.*⌽-⎕IO-⍳⍴X                       X←D1; Y←D1; G←D0
```
Interpolated value of series (X,Y) at G

```
68  ' *○⍟'[⎕IO+2⊥X∘.≥⍳⌈/,X]                    X←I2; 1⍴⍴X ↔ 2
```
Barchart of two integer series (across the page)

69 `(X∘.+,0)⊥Y` X←D1; Y←D1
Value of polynomial with coefficients Y at points X

70 `((A⍳A)=⍳⍴A←2⊥X∧.=⍉X)⌿X` X←A2
Removing duplicate rows

71 `B⍴A ⋄ A[⎕IO+B[1]⊥-⎕IO-X]←1 ⋄ A←(×/B←0 0+⌈/,X)⍴0` X←C2
Transforming connectivity list X into a connectivity matrix

72 `100⊥100|3↑⎕TS`
Encoding current date

73 `(1-(' '=X)⊥1)↓X` X←C1
Removing trailing blanks

74 `(12⍴7⍴31 30)[X]-0⌈¯1+2⊥(X=2),[.1]0≠4|Y` X←I0; Y←I
Number of days in month X of year Y

75 `16⊥-⎕IO-'0123456789ABCDEF'⍳⍉X` X←C
Conversion from hexadecimal to decimal

76 `(1-(' '=X)⊥1)⌽X` X←C
Justifying right

77 `(÷1+Y÷100)⊥⌽X` X←D1; Y←D0
Present value of cash flows X at interest rate Y %

78 `10⊥¯1+'0123456789'⍳X` X←C1
Transformation of alphanumeric string into numeric

79 `(' '=X)⊥1` X←C1
Index of first non-blank, counted from the rear

80 `(,X)[⎕IO+(⍴X)⊥Y-⎕IO]` X←A; Y←I2
Indexing scattered elements

81 `⎕IO+(⍴X)⊥Y-⎕IO` X←A; Y←I2
Conversion of indices to indices of raveled array

82 `(1+Y÷100)⊥X` X←D1; Y←D0
Future value of cash flows X at interest rate Y %

83 `X⊥Y` X←D0; Y←D
Value of polynomial with coefficients Y at point X

84 `2⊥X` X←B
Integer representation of logical vectors

EXECUTE ⍎

85 `⍎⍎'1','↑↓'[⎕IO+∧/(⍴X)=⌽⍴X],'''0~0∊X=-⍉X'''` X←D2
Test for antisymmetricity of matrix X

```
 86  ⍎⍎'1','↑↓'[⎕IO+∧/(ρX)=⌽ρX],'''0~0∊X=⍉X'''                   X←A2
     Test for symmetricity of matrix X

 87  ⌊10⍟(⍎('.'≠A)/A←⍕X)÷X                                       X←D1
     Number of decimals of elements of X

 88  ⍎'VAR',(⍕X),'←Y'                                      X←A0; Y←A
     Using a variable named according to X

 89  ⍎X/'→'                                                      X←L0
     Conditional branch out of programs

 90  ' *'[⎕IO+(⌽(¯1+⌊/A)+⍳1+(⌈/A)-⌊/A)∘.=A←⌊.5+⍎F]     F←A1; X←D1
     Graph of F(X) at points X ('X'∊F)

 91  A+.×⍎F,0ρX←Y[1]+(A←--/Y÷G)×0,⍳G    F←A1; G←D0; Y←D1; ρY ↔ 2
     Definite integral of F(X) in range Y with G steps ('X'∊F)

 92  X←⍎,((2↑'X'),' ',[.5]A)[⎕IO+~' '∧.=A←,⍞;]
     Changing X if a new input value is given

 93  (X∨.≠' ')\1↓⍎'0 ',,X,' '                                    X←C2
     Conversion of each row into a number (default zero)

 94  ⍎((X∧.=' ')/'Y'),X                                          X←D1
     Converting expression X to numeric form with default value Y

 95  1↓⍎'0 ',(∧/X∊' 0123456789')/X                               X←C1
     Test if numeric and converting into numeric form

 96  ⍎X/'EXPRESSION'                                             X←B0
     Conditional execution

 97  1ρ(⍎⍞,',⍳0'),X                                               X←D0
     Giving a numeric default value for input

 98  A←⍎,',','(','0','ρ',Y,'←',X,')'                      X←C2; Y←C2
     Assign values of expressions in X to variables named in Y

 99  ⍎,',','(',',',X,')'                                          X←A
     Evaluation of several expressions; results form a vector

100  ⍎'X[',((¯1+ρρX)ρ';'),'Y]'                              X←A; Y←I
     Indexing when rank is not known beforehand

101  ⍎,'+',X                                                     X←A2
     Sum of numbers in character matrix X

---
```

FORMAT ⍕

```
102  (3⌽7 0⍕X∘.+,0),⍕Y                                    X←D1; Y←A2
     Numeric headers (elements of X) for rows of table Y
```

```
103  ⍕X∘.+,0                                                 X←D1
     Formatting a numerical vector to run down the page

104  A Δ A[(' '=A)/⍳⍴A]←'.' Δ A←⍕⌽3↑⎕TS
     Representation of current date (ascending format)

105  (,(0 1↓3 0⍕100+3 1⍴12 0 0|3↑3↓⎕TS),':: '),'AP'[1+12≤⎕TS[4]],'M'
     Representation of current time (12 hour clock)

106  0 1↓(2↑Y+1)⍕(10*Y)+((⍴X),1)⍴X                     X←I1; Y←I0
     Leading zeroes for positive numbers X in fields of width Y

107  A Δ A[(' '=A)/⍳⍴A]←'/' Δ A←⍕100|1⌽3↑⎕TS
     Representation of current date (American)

108  (⍴A)⍴B\(B←,('0'≠A)∨' '≠¯1⌽A)/,A←' ',⍕X                   X←A
     Formatting with zero values replaced by blanks

109  ((1,G)×⍴X)⍴2 1 3⍉(⌽G,⍴X)⍴(,G,[1.1]Y)⍕⍉X      X←D2; Y←I1; G←I0
     Row-by-row formatting (width G) of X with Y decimals per row

110  (,G,[1.1]H)⍕X                                X←D; G←I1; H←I1
     Formatting X with H decimals in fields of width G

111  ⍴⍕X                                                     X←D0
     Number of digit positions in scalar X (depends on ⎕PP)

---
     BRANCH  →

112  →⎕LC⌈⍳∨/,(X←X∨X∨.∧X)≠+X                                X←B2
     Forming a transitive closure

113  →Y⌈⍳G≥X←X+1                                  X←I0; Y←I0; G←I0
     For-loop ending construct

114  →Y⌈⍳X                                               X←B0; Y←I0
     Conditional branch to line Y

115  →X/Y                                                 X←B1; Y←I1
     Case structure with logical switch (preferring from start)

116  →X⌽Y                                                 X←I0; Y←I1
     Case structure with integer switch

117  →Y[G⍳X]                                      X←A0; Y←I1; G←A1
     Case structure according to key vector G

118  →0×⍳X                                                    X←B0
     Conditional branch out of program

119  →Y[2+×X]                                           . X←I0; Y←I1
     Conditional branch depending on sign of X
```

```
120  →Y××X                                                  X←D0; Y←I0
     Continuing from line Y (if X>0) or exit

---
     ROLL / DEAL  ?

121  X[1]+?Yρ--/X                                           X←I1; Y←I1
     Y-shaped array of random numbers within ( X[1],X[2] ]

122  ?YρX                                                     Y←I; X←I
     Choosing Y objects out of ⍳X with replacement (roll)

123  Y?X                                                    X←I0; Y←I0
     Choosing Y objects out of ⍳X without replacement (deal)

---
     MATRIX INVERSION / MATRIX DIVISION  ⌹

124  *A+.×(⍟Y)⌹A←X∘.*0 1                                   X←D1; Y←D1
     Predicted values of exponential (curve) fit

125  A+.×Y⌹A←X∘.*0 1                                       X←D1; Y←D1
     Predicted values of best linear fit (least squares)

126  ⌽Y⌹X∘.*0,⍳G                                           X←D1; Y←D1
     G-degree polynomial (curve) fit of points (X,Y)

127  A ⍝ A[1]←*A[1] ⍝ A←(⍟Y)⌹X∘.*0 1                       X←D1; Y←D1
     Cofficients of exponential (curve) fit of points (X,Y)

128  Y⌹X∘.*0 1                                             X←D1; Y←D1
     Best linear fit of points (X,Y) (least squares)

---
     GEOMETRICAL  ○

129  ((X≠0)×¯3○Y÷X+X=0)+○((X=0)×.5××Y)+(X<0)×1-2×Y<0      X←D; Y←D
     Arctan Y÷X

130  1 2×.○X,Y                                             X←D0, Y←D0
     A way to combine trigonometric functions (sin X cos Y)

131  ○/¯2 1,X                                                    X←D0
     Complementary angle (arccos sin X)

132  2 2ρ1 ¯1 1 1×2 1 1 2○X                                      X←D0
     Rotation matrix for angle X (in radians) counter-clockwise

133  X×180÷○1                                                     X←D
     Transformation from radians to degrees

134  X×○÷180                                                      X←D
     Transformation from degrees to radians
```

```
---
     FACTORIAL / BINOMIAL  !

135  (!Y)×Y!X                                                 X←D; Y←D
     Number of permutations of X objects taken Y at a time

136  ⍉A∘.!A←0,⍳X                                                  X←I0
     Pascal's triangle of order X (binomial coefficients)

137  +/Y×(X*A)÷!A←¯1+⍳⍴Y                                     X←D0; Y←D1
     Value of Taylor series with coefficients Y at point X

138  (A!X)×(Y*A)×(1-Y)*X-A←-⎕IO-⍳X+1                         X←I0; Y←D0
     Binomial distribution of X trials with probability Y

139  ÷Y×(X-1)!Y+X-1                                          X←D0; Y←D0
     Beta function

140  (*-Y)×(Y*X)÷!X                                           X←I; Y←D0
     Poisson distribution of states X with average number Y

141  !X-1                                                         X←D0
     Gamma function

142  Y!X                                                      X←D; Y←D
     Number of combinations of X objects taken Y at a time

---
     OUTER PRODUCT  ∘.ω

143  ⍉X∘.=(1 1⍉<\X∘.=X)/X                                         X←D1
     'Position' matrix of a vector

144  ' ⎕'[⎕IO+(⌽⍳⌈/A)∘.≤A←+/(⍳1+(⌈/X)-⌊/X)∘.=X]                   X←I1
     Histogram (distribution barchart; down the page)

145  +/((¯1↓Y)∘.≤X)∧(1↓Y)∘.>X                                 X←D; Y←D1
     Distribution of X into intervals between Y

146  Y∘.×(1+G÷100)∘.*X                                   X←D; Y←D; G←D
     Compound interest for principals Y at rates G % in times X

147  (Y[A∘.+¯1+⍳⍴X]∧.=X)/A←(A=1↑X)/⍳⍴A←(1-⍴X)↓Y           X←A1; Y←A1
     Occurrences of string X in string Y

148  (1 ¯1∘.=⍉X)+.×⍳1↑⍴X                      X←I2 (1=START; ¯1=END)
     Transforming connection matrix X into a node matrix

149  0+.≠(⌈(10*Y)×10*⎕IO-⍳Y+1)∘.|⌈X×10*Y                     X←D; Y←I0
     Number of decimals (up to Y) of elements of X

150  X+.×Y∘.=((⍳⍴Y)=Y⍳Y)/Y                                   X←D1 Y←D1
     Sum over elements of X determined by elements of Y
```

```
151  (F A)+.×(A←((⍳⍴X)=X⍳X)/X)∘.=X                              X←A1
     Executing costly monadic function F on repetitive arguments

152  (G∘.=X)+.×Y                                      X←A; Y←D; G←A
     Sums according to codes G

153  1++/∧\1 2 1 3⍉Y∘.≠X                                  X←A2; Y←A2
     'X⍳Y' by rows for elements of matrices

154  (1 1⍉<\X∘.=X)/X                                             X←A1
     Removing duplicate elements (nub)

155  ⌈/(∧/0=A∘.∣X)/A←⍳⌊/X                                        X←I1
     Greatest common divisor of elements of X

156  +/(A×X÷⌈/A←Y-⌊/Y)∘.≥¯1+⍳X                            X←I0; Y←D1
     Classification of elements Y into X classes of equal size

157  -/(⍳⌈/,X)∘.=⍉X                    X←I2 (X[1;]=START; X[2;]=END)
     Transforming node matrix X into a connection matrix

158  ∧/1=+/X∘.=⍳⍴X                                               X←I1
     Test if X is a permutation vector

159  ∨/∧/0 1∘.=X                                                 X←B1
     Test if all elements of vector X are equal

160  (⍴X)⍴(,(+/A)∘.>-⎕IO-⍳¯1↑⍴X)\(,A←X≠' ')/,X                    X←C
     Moving all blanks to end of each row

161  ∧/,(0≠X)≤A∘.≤A←⍳1↑⍴X                                        X←D2
     Test if X is an upper triangular matrix

162  ∧/,(0≠X)≤A∘.≥A←⍳1↑⍴X                                        X←D2
     Test if X is a lower triangular matrix

163  +⌿(⎕IO-⍳⍴X)⌽X∘.×Y,0×1↓X                              X←D1; Y←D1
     Product of two polynomials with coefficients X and Y

164  0=(⍳⌈/X)∘.∣X                                                X←I1
     Divisibility table

165  X∘.<⌽⍳⌈/X                                                   X←I1
     Matrix with X[i] trailing zeroes on row i

166  ' ⎕'[⎕IO+(⌽⍳⌈/X)∘.≤X]                                       X←I1
     Barchart of integer values (down the page)

167  X∘.≥⌽⍳⌈/X                                                   X←I1
     Matrix with X[i] trailing ones on row i

168  X∘.<⍳⌈/X                                                    X←I1
     Matrix with X[i] leading zeroes on row i
```

169 `X∘.≥⍳⌈/X,0` X←I1
Comparison table

170 `' ⎕'[⎕IO+X∘.≥(⌈/X)×(⍳Y)÷Y]` X←D1; Y←D0
Barchart of X with height Y (across the page)

171 `' ⎕'[⎕IO+X∘.≥⍳⌈/X]` X←I1
Barchart of integer values (across the page)

172 `X∘.≥⍳⌈/X` X←I1
Matrix with X[i] leading ones on row i

173 `+/(⍳Y)∘.=⌈(X-G)÷H` X←D; Y←I0; G←D0; H←D0
Division to Y classes with width H, minimum G

174 `1 2 1⍉X∘.-⌊/X` X←D2
Move set of points X into first quadrant

175 `(2=+⌿0=(⍳X)∘.|⍳X)/⍳X` X←I0
All primes up to X

176 `(∧⌿(¯1+⍳⍴X)⌽(X∘.=Y),0)/⍳1+⍴Y` X←C1; Y←C1
Occurrences of string X in string Y

177 `⎕IO++/Y∘.≥(' '=X)/⍳⍴X` X←C1; Y←I
Ordinal numbers of words in X that indices Y point to

178 `(∧⌿(¯1+⍳⍴X)⌽X∘.=Y)⍳1` X←A1; Y←A1
First occurrence of string X in string Y

179 `Y[+⌿Y∘.≤X]` X←D0; Y←D1
Contour levels Y at points with altitudes X

180 `≠/X∘.≥Y` X←D; Y←D1
Test if X is within range [Y[1],Y[2])

181 `+/X∘.≥0 50 100 1000` X←D
Which class do elements of X belong to

182 `(((¯1⌽~A)∧A←(¯1↓X=1⌽X),0)/Y)∘.=Y` X←A1; Y←A1
Repeat matrix

183 `(⍳X)∘.⌈⍳X` X←I0
Maximum table

184 `(,Y∘.>⌽(⍳G)-⎕IO)\X` X←A1; Y←I1; G←I0
Justifying right fields of X (lengths Y) to length G

185 `(,Y∘.>(⍳G)-⎕IO)\X` X←A1; Y←I1; G←I0
Justifying left fields of X (lengths Y) to length G

186 `((⍴A)⍴Y÷100)÷A←⍉1-(1+Y÷100)∘.*-X` X←I; Y←D
Annuity coefficient: X periods at interest rate Y %

```
187  1 3 2 4⍉X∘.×Y                                   X←D2; Y←D2
     Direct matrix product

188  1 2 1 2⍉X∘.×Y                                   X←D2; Y←D2
     Shur product

189  1 2 2⍉Y∘.+X                                     X←D1; Y←D2
     Adding X to each row of Y

190  2 1 2⍉X∘.+Y                                     X←D1; Y←D2
     Adding X to each row of Y

191  1 2 1 2⍉X∘.+Y                                   X←D2; Y←D2
     Matrix sum

192  1 2 1⍉Y∘.+X                                     X←D1; Y←D2
     Adding X to each column of Y

193  1 1 2⍉X∘.+Y                                     X←D1; Y←D2
     Adding X to each column of Y

194  G⍉X∘.αY                                   X←A; Y←A; G←I1
     Selecting specific elements from a 'large' outer product

195  (⍳X)∘.≤⍳X                                              X←I0
     X×X upper triangular matrix

196  (⍳X)∘.≥⍳X                                              X←I0
     X×X lower triangular matrix

197  (⍳X)∘.=⍳X                                              X←I0
     X×X identity matrix

198  ÷¯1+(⍳X)∘.+⍳X                                          X←⍳0
     Hilbert matrix of order X

199  (⍳X)∘.×⍳X                                              X←I0
     Multiplication table

200  X[;,(Y⍴1)∘.×⍳(⍴X)[2];]                          Y←I0; X←A3
     Replicating a dimension of rank three array X Y-fold

201  (0,⍳(⍴X)-Y)∘.+Y                                   X←A1 Y←I0
     Moving index of width Y for vector X

202  X∘.+⍳Y                                          X←I1; Y←I0
     Indices of subvectors of length Y starting at X+1

203  X∘.+,0                                                 X←D1
     Reshaping numeric vector X into a one-column matrix

204  X∘.×1 ¯1                                                X←D
     Array and its negative ('plus minus')
```

INNER PRODUCT α.ω

```
205  (-2↑+/∧\⌽X∧.=' ')↓X                                      X←C2
     Removing trailing blank rows

206  ((A⍳A)=⍳⍴A←⎕IO++⌿∧⍀X∨.≠⍉X)⌿X                            X←A2
     Removing duplicate rows

207  ⎕IO++⌿∧⍀X∨.≠⍉Y                                    X←A2; Y←A2
     'X⍳Y' for rows of matrices

208  (1 1⍉<\X∧.=⍉X)⌿X                                         X←A2
     Removing duplicate rows

209  (⌽∨\⌽' '∨.≠X)/X                                          X←C2
     Removing trailing blank columns

210  (∨\' '∨.≠X)/X                                            X←C2
     Removing leading blank columns

211  (∨\X∨.≠' ')⌿X                                            X←C2
     Removing leading blank rows

212  (X⌈1↓A)⌊1↑A←(2 2⍴¯1 1 1 ¯.1)+.×10*(-1↓Y),-/Y+Y>99 0 X←D; Y←I1
     Limits X to fit in ⍕ field Y[1 2]

213  A+(X-A←⌊/X)⌈.×Y                                     X←A1; Y←B
     Maxima of elements of subsets of X specified by Y

214  X∧.=∨/X                                                  X←B1
     Test if all elements of vector X are equal

215  X∧.=∧/X                                                  X←B1
     Test if all elements of vector X are equal

216  ((((1↑⍴X),⍴Y)↑X)∧.=Y)⌿X                           X←A2; Y←A1
     Rows of matrix X starting with string Y

217  (' '≠X)⌈.×⍳¯1↑⍴X                                          X←C
     Indices of last non-blanks in rows

218  (A∨1↓1⌽1,A←X∨.≠' ')⌿X                                    X←C2
     Removing duplicate blank rows

219  ((-A)↓X∧.=(A,1+⍴Y)⍴Y)/⍳(⍴Y)+1-A←⍴X               X←A1; Y←A1
     Occurrences of string X in string Y

220  (A∨1,¯1↓A←' '∨.≠X)/X                                     X←C2
     Removing duplicate blank columns

221  ~X≤.≥(⌈X),G,H                               X←I0; G←I0; H←I0
     Test if X is an integer within range [ G,H )
```

```
222  Y⌈.×X                                        X←D1; Y←D1
     Maximum of X with weights Y

223  Y⌊.×X                                        X←D1; Y←D1
     Minimum of X with weights Y

224  X←X⌊.+X                                            X←B2
     Extending a distance table to next leg

225  (X∨.≠' ')⌿X                                        X←C2
     Removing blank rows

226  (' '∨.≠X)/X                                        X←C2
     Removing blank columns

227  (X*¯1+⍳⍴Y)+.×⌽Y                               X←D0; Y←D
     Value of polynomial with coefficients Y at point X

228  1∊X∧.=Y                                       X←A; Y←A1
     Test if vector Y is a row of array X

229  (Y∧.=X)⍳1                         X←A2; Y←A1; ¯1↑⍴Y↔⍴X
     First occurrence of string X in matrix Y

230  X←X∨.∧X                                            X←B2
     Extending a transitive binary relation

231  X∨.≠Y                                         X←A; Y←A0
     Test if rows of X contain elements differing from Y

232  X∧.=Y                                         X←A; Y←A1
     Comparing vector Y with rows of array X

233  X<.<Y                            X←D0; Y←D2; 1↑⍴Y ↔ 2
     Test if X is within range [ Y[1;],Y[2;] )

234  X<.≤Y                             X←D; Y←D2; 1↑⍴Y ↔ 2
     Test if X is within range ( Y[1;],Y[2;] ]

235  (X,[.1+⍴⍴X]X)>.>Y                 X←D; Y←D2; 1↑⍴Y ↔ 2
     Test if X is within range ( Y[1;],Y[2;] ]

236  X+.=,Y                                        X←A0; Y←A
     Number of occurrences of scalar X in array Y

237  (Y+.×X)÷⍴X                                   X←D1; Y←D1
     Arithmetic average (mean value) of X weighted by Y

238  Y-.÷X                                        X←D1; Y←D1
     Sum of alternating reciprocal series Y÷X

239  Y+.÷X                                        X←D1; Y←D1
     Sum of reciprocal series Y÷X
```

```
240  X+.×Y                                  X←D; Y←D; ¯1↑ρX ↔ 1↑ρY
     Matrix product

241  X+.×Y                                                X←A1; Y←B
     Summation over subsets of X specified by Y

242  X+.×X                                                     X←D1
     Sum of squares of X

243  Y+.×X                                               X←D1; Y←D1
     Scalar (dot) product of vectors

244  X×.*Y                                                X←A1; Y←B
     Products over subsets of X specified by Y

245  ⎕RL←⎕TS+.*2
     Randomizing random numbers (in ⎕LX in a workspace)

246  X+.*2                                                     X←D1
     Sum of squares of X

---
     SCAN  ω\ ω⍀

247  (⍳+/X,Y)∊+\1+¯1↓0,((⍳+/X)∊+\X)\Y                        Q←I1; Y←I1
     Vector (X[1]ρ1),(Y[1]ρ0),(X[2]ρ1),...

248  (A-⌊0.5×(A←+/∧\⌽A)++/∧\A←' '=⌽X)⌽X                            X←C
     Centering character array X with ragged edges

249  ⎕IO++\1+((⍳+/X)∊+\⎕IO,X)\Y-¯1↓1,X+Y         X←I1; Y←I1; ρX↔ρY
     Vector (Y[1]+⍳X[1]),(Y[2]+⍳X[2]),(Y[3]+⍳X[3]),...

250  Y[+\(⍳+/X)∊¯1↓1++\0,X]                      X←I1; Y←A1; ∧/0<X
     Replicate Y[i] X[i] times (for all i)

251  ((X≠0)/Y)[+\¯1⌽(⍳+/X)∊+\X]                             X←I1; Y←A1
     Replicate Y[i] X[i] times (for all i)

252  ≠\(⍳+/X)∊+\⎕IO,X                                     X←I1; ∧/0<X
     Vector (X[1]ρ1),(X[2]ρ0),(X[3]ρ1),...

253  Y[⎕IO++\(⍳+/X)∊⎕IO++\X]                     X←I1; Y←A1; ∧/0<X
     Replicate Y[i] X[i] times (for all i)

254  ≠\Y≠X\A≠¯1↓0,A←X/≠\¯1↓0,Y                               X←B1; Y←B1
     Running parity (≠\) over subvectors of Y indicated by X

255  +\Y-X\A-¯1↓0,A←X/+\¯1↓0,Y                               X←B1; Y←D1
     Cumulative sum (+\) over subvectors of Y indicated by X

256  Y∧A=⌈\X×A←+\Y>¯1↓0,Y                                     X←B; Y←B
     Groups of ones in Y pointed by X (or trailing parts)
```

```
257  A-¯1↓0,A←(+\X)[+\Y]                             X←D1; Y←I1
     Sums over (+/) subvectors of X, lengths in Y

258  (⍴X)⍴A\(A←,∧\('⍝'≠X)∨≠\X='''')/,X                      X←C2
     Decommenting a matrix representation of a function (⎕CR)

259  ((⌽∨\⌽A)∧∨\A←' '≠X)/X                                  X←C1
     Removing leading and trailing blanks

260  +\+\⍳X                                                  X←I0
     X first figurate numbers

261  X∧∧\X=∨\X                                                X←B
     First group of ones

262  (<\,(X=(⍴X)⍴⌈⌿X)∧X=⍉(⌽⍴X)⍴⌊/X)/,X                      X←D2
     Value of saddle point

263  ((~(⍴A↑X)↑'/'=Y)/A↑X),(1↓A↓Y),(A←+/∧\Y≠',')↓X    X←C1; Y←C1
     Editing X with Y ∇-wise

264  ((⍳(⍴Y)++/X)∊+\1+0,¯1↓((⍳⍴Y)∊G)\X)\Y       X←I1; Y←A1; G←I1
     Open a gap of X[i] after Y[G[i]] (for all i)

265  (⍳(⍴Y)++/X)∊+\1+¯1↓0,(1⌽Y)\X                     X←I1; Y←B1
     Insert vector for X[i] zeroes after i:th subvector

266  (-+/∧\' '=⌽X)↓X                                        X←C1
     Removing trailing blanks

267  (+/∧\' '=X)↓X                                          X←C1
     Removing leading blanks

268  ∧/[Y]X=⌈\[Y]X                                       X←D; Y←I0
     Test if X is in ascending order along direction Y

269  (-⌊0.5×+/∧\' '=⌽X)⌽X                                    X←C
     Centering character array X with only right edge ragged

270  ∧/[Y]X=⌊\[Y]X                                       X←D; Y←I0
     Test if X is in descending order along direction Y

271  ((⍳(⍴Y)++/X)∊+\1+((⍳⍴Y)∊G)\X)\Y             X←I1; Y←A1; G←I1
     Open a gap of X[i] before Y[G[i]] (for all i)

272  (⌽∨\⌽∨⌿' '≠X)/X                                        X←C2
     Removing trailing blank columns

273  X,[⎕IO-(⍲\2⍴∧/2>(⍴⍴X),⍴⍴Y)/.5 0]Y                   X←A; Y←A
     Vectors as matrices in catenation

274  X,[⎕IO+(⍲\2⍴∧/2>(⍴⍴X),⍴⍴Y)/.5 1]Y                   X←A; Y←A
     Vectors as matrices in catenation
```

```
275  (-+/∧\' '=⌽X)⌽X                                                X←C
     Justifying right

276  (+/∧\' '=X)⌽X                                                  X←C
     Justifying left

277  (⍳+/X)∊(+\X)-~⎕IO                                             X←I1
     Changing lengths of subvectors to ending indicators

278  (⍳+/X)∊+\⎕IO,X                                                X←I1
     Changing lengths of subvectors to starting indicators

279  (⍳+/A)∊+\A←1+X                                                X←I1
     Insert vector for X[i] elements before i:th element

280  (+\X)⍳⍳+/X                                                    X←B1
     Indices of ones in logical vector X

281  +/Y××\1,X÷⍳¯1+⍴Y                                        X←D0; Y←D1
     Value of Taylor series with coefficients Y at point X

282  ⎕IO++/∧\' '≠X                                                  X←C
     Indices of first blanks in rows of array X

283  (Y=+\X=1↑X)/X                                           Y←I0; X←C1
     Locating field number Y starting with first element of X

284  A-¯1↓0,A←(Y≠1↓Y,0)/+\X                                  X←D1; Y←D1
     Sum elements of X marked by succeeding identicals in Y

285  ((Y-1)↓A)-0,(-Y)↓A←+\X                                  X←D1; Y←I0
     Running sum of Y consecutive elements of X

286  G-¯1↓0,G←0⌈(+\Y)-X                                      Y←D1; X←D0
     Fifo stock Y decremented with X units

287  A∨¯1↓0,A←2|+\X=''''                                           X←C1
     Locations of texts between and including quotes

288  A∧¯1↓0,A←2|+\X=''''                                           X←C1
     Locations of texts between quotes

289  ≠\(Y∨X)\A≠¯1↓0,A←(Y∨X)/Y                                X←B1; Y←B1
     Or-scan (∨\) over subvectors of Y indicated by X

290  ~≠\(Y≤X)\A≠¯1↓0,A←~(Y≤X)/Y                              X←B1; Y←B1
     And-scan (∧\) over subvectors of Y indicated by X

291  A-¯1↓0,A←(1⌽X)/+\Y                                      X←B1; Y←D1
     Sums over (+/) subvectors of Y indicated by X

292  Y∧A∊(X∧Y)/A←+\Y>¯1↓0,Y                                  X←B1; Y←B1
     Groups of ones in Y pointed by X
```

```
293  A∨¯1↓0,A←≠\X=''''                                     X←C1
     Locations of texts between and including quotes

294  A∧¯1↓0,A←≠\X=''''                                     X←C1
     Locations of texts between quotes

295  +\('('=X)-¯1↓0,')'=X                                  X←C1
     Depth of parentheses

296  +\¯1↓⎕IO,X                                            X←I1
     Starting positions of subvectors (lengths in X)

297  X[1⌈⌈\Y×⍳⍴Y]                                    X←A1; Y←B1
     Duplicating element of X belonging to Y,1↑X until next found

298  (⌽∨\⌽' '≠X)/X                                         X←C1
     Removing trailing blanks

299  (∨\' '≠X)/X                                           X←C1
     Removing leading blanks

300  (G=+\X)/Y                                  X←A1; Y←B1; G←I0
     G:th subvector of Y (subvectors indicated by X)

301  -\⍳X                                                  X←I0
     Alternating series (1 ¯1 2 ¯2 3 ¯3 ...)

302  +\⍳X                                                  X←I0
     X first triangular numbers

303  X∨≠\X                                                  X←B
     Joining pairs of ones

304  ∨\X                                                    X←B
     Turn on all zeroes after first one

305  (~X)∧≠\X                                               X←B
     Places between pairs of ones

306  ∧\X                                                    X←B
     Turn off all ones after first zero

307  <\X                                                    X←B
     Turn off all ones after first one

308  ≤\X                                                    X←B
     Turn on all zeroes after first zero

309  ≠\X                                                    X←B
     Running parity

310  +\X                                                    X←D
     Cumulative sum
```

REDUCTION ω/ ω⌿

```
311  (⌈/X)=⌊/X                                             X←D1
     Test if all elements of vector X are equal

312  (⌈/X)-⌊/X                                             X←D1
     Size of range of elements of X

313  -/×⌿0 1⌽X                                             X←D2
     Evaluating a two-row determinant

314  -/×/0 1⊖X                                             X←D2
     Evaluating a two-row determinant

315  (∧/X)∨~∨/X                                            X←B1
     Test if all elements of vector X are equal

316  (∧/X)=∨/X                                             X←B1
     Test if all elements of vector X are equal

317  ∧/X÷∨/X                                               X←B1
     Test if all elements of vector X are equal

318  (×/(+/X÷2)-0,X)*.5                           X←D1; 3 ↔ ρX
     Area of triangle with side lengths in X (Heron's formula)

319  ((+/(X-(+/X)÷ρX)*2)÷ρX)*.5                            X←D1
     Standard deviation of X

320  (+/(X-(+/X)÷ρX)*2)÷ρX                                 X←D1
     Variance (dispersion) of X

321  (+/(X-(+/X)÷ρX)*Y)÷ρX                                 X←D1
     Y:th moment of X

322  ∧/((1↑X)∊10↓A),X∊A←'0..9A..ZΔA̲..X̲Δ̲'                     X←C1
     Test if X is a valid APL name

323  Y↑((⌊-/.5×Y,ρX)ρ' '),X                          X←C1; Y←I0
     Centering text line X into a field of width Y

324  ∧/X=1↑X                                               X←A1
     Test if all elements of vector X are equal

325  (+/X)÷¯1↑1,ρX                                         X←D2
     Average (mean value) of rows of matrix X

326  (+⌿X)÷1↑(ρX),1                                        X←D2
     Average (mean value) of columns of matrix X

327  (¯1⌽1↓(∨/X≠¯1⊖X),1)⌿X                                 X←A2
     Removing duplicate rows from ordered matrix X
```

328 `×/¯1↓⍴X` X←A
Number of rows in array X (also of a vector)

329 `(⍳⌈/X)∊X` X←I1
Converting set of positive integers X into a mask

330 `X⍳⌈/X` X←D1
Index of (first) maximum element of X

331 `⌈/⍳0`
Negative infinity; the smallest representable value

332 `(+/,X)÷1⌈⍴,X` X←D
Arithmetic average (mean value), also for an empty array

333 `A[X] ∆ A[Y]←1 ∆ A←(⌈/X,Y)⍴0` X←I1; Y←I1
Quick membership (∊) for positive integers

334 `⌈/X,0` X←D1
Positive maximum, at least zero (also for empty X)

335 `⌈/X` X←D1
Maximum of elements of X

336 `X⍳⌊/X` X←D1
Index of (first) minimum element of X

337 `⌊/⍳0`
Positive infinity; the largest representable value

338 `⌊/X⍳Y` X←C1; Y←C1
Index of first occurrence of elements of Y

339 `⌊/X` X←D1
Minimum of elements of X

340 `0=(⍴X)|+/X` X←B1
Test if all elements of vector X are equal

341 `∧/X/1⌽X` X←B1
Test if all elements of vector X are equal

342 `+/A×¯1*A<1⌽A←0,1000 500 100 50 10 5 1['MDCLXVI'⍳X]` X←A
Interpretation of roman numbers

343 `∧/X=1⌽X` X←A1
Test if all elements of vector X are equal

344 `∧/X=1⊖X` X←A2
Comparison of successive rows

345 `∧/(X∊Y),Y∊X` X←A1; Y←A1
Identity of two sets

346 `∧/(⍳⍴X)∊X` X←I1
Test if X is a permutation vector

347 `⍲/0 1∊X` X←B1
Test if all elements of vector X are equal

348 `∨/Y∊X` X←A; Y←A1
Test if X and Y have elements in common

349 `~∧/X∊~X` X←B1
Test if all elements of vector X are equal

350 `∧/,X∊0 1` X←A
Test if X is boolean

351 `∧/Y∊X` X←A; Y←A1
Test if Y is a subset of X (Y ⊂ X)

352 `≠/0 1∊X` X←B1
Test if all elements of vector X are equal

353 `∧/(X⍳X)=⍳⍴X` X←A1
Test if each element of X occurs only once

354 `∧/⎕IO=X⍳X` X←A1
Test if all elements of vector X are equal

355 `~∨/X` X←B
None, neither

356 `∨/X` X←B
Any, anyone

357 `∧/,X=Y` X←A; Y←A; ⍴X ↔ ⍴Y
Test if arrays of equal shape are identical

358 `∧/X=X[1]` X←A1
Test if all elements of vector X are equal

359 `∧/' '=X` X←C2
Blank rows

360 `∧/X` X←B
All, both

361 `≠/X` X←B
Parity

362 `+/X=,Y` X←A0; Y←A
Number of occurrences of scalar X in array Y

363 `(-X[2]-¯1 1×((X[2]*2)-×/4,X[1 3])*.5)÷2×X[1]` X←D1; 3 ↔ ⍴X
(Real) solution of quadratic equation with coefficients X

```
364  (+/[Y]X)÷(ρX)[Y]                                X←D; Y←I0
     Average (mean value) of elements of X along direction Y

365  (+/X)÷ρX                                              X←D1
     Arithmetic average (mean value)

366  ×/ρX                                                   X←A
     Number of elements (also of a scalar)

367  ÷/X                                                    X←D
     Alternating product

368  ×/X                                                   X←D1
     Product of elements of X

369  -/X                                                    X←D
     Alternating sum

370  +/X                                                   X←B1
     Number of elements satisfying condition X

371  +/X                                                   X←A1
     Reshaping one-element vector X into a scalar

372  +⌿X                                                   X←D2
     Column sum of a matrix

373  +/X                                                   X←D2
     Row sum of a matrix

374  +/X                                                   X←D1
     Sum of elements of X

---
     TAKE  ↑

375  Y[⍳G;],[1]((1↓ρY)↑X),[1](2↑G)↓Y         X←A1; Y←A2; G←I0
     Inserting vector X into matrix Y after row G

376  (A↑X),[⍳1](1↓A←(ρX)⌈0,ρY)↑Y                    X←A2; Y←A1
     Inserting vector Y to the end of matrix X

377  Y↑X,Yρ¯1↑X                                     X←A1; Y←I0
     Filling X with last element of X to length Y

378  X[Y;]←(1↑ρX)↑⍞                                 X←C2; Y←I0
     Input of row Y of text matrix X

379  (1↑A)↓(A⍲1⌽A←X=Y)/X                            X←A1; Y←A0
     Removing leading, multiple and trailing Y's

380  X←(ρX)ρA ∆ A[(A=1↑Y)/⍳ρA←,X]←1↓Y                X←A; Y←A1
     Changing elements in X with value Y[1] into Y[2]
```

381 `X>((-ρρX)↑¯1)↓0,X` `X←B`
First ones in groups of ones

382 `(G↑Y),X,G↓Y` `X←A1; Y←A1; G←I0`
Inserting X into Y after index G

383 `X-((-ρρX)↑¯1)↓0,X` `X←D`
Pairwise differences of successive columns (inverse of +\)

384 `((-ρρX)↑1)↓X,0` `X←D`
Rightmost neighboring elements

385 `((-ρρX)↑¯1)↓0,X` `X←D`
Leftmost neighboring elements

386 `(-ρX)↑(-Y)↓X` `X←A1; Y←I0`
Shifting vector X right with Y without rotate

387 `(ρX)↑Y↓X` `X←A1; Y←I0`
Shifting vector X left with Y without rotate

388 `(2↑Y)↓X` `X←A2; Y←I0`
Drop of Y first rows from matrix X

389 `,⍉(Aρ2)ρ(2*A←⌈2⍟X)↑⍳X` `X←I0`
Playing order in a cup for X ranked players

390 `((1 0×ρY)⌈ρX)↑X` `X←A2; Y←A2`
Lengthening matrix X to be compatible with Y

391 `((0 1×ρY)⌈ρX)↑X` `X←A2; Y←A2`
Widening matrix X to be compatible with Y

392 `(1⌈¯2↑ρX)ρX` `X←A; 2≥ρρX`
Reshaping non-empty lower-rank array X into a matrix

393 `(-⌊.5×Y+ρX)↑X` `X←C1; Y←I0`
Centering text line X into a field of width Y

394 `(X⌊ρY)↑Y` `X←I; Y←A`
Take of at most X elements from Y

395 `(¯1↑ρY)|(,Y)⍳X` `Y←C2; X←C`
Alphabetizing X; equal alphabets in same column of Y

396 `(~(⍳¯1↑ρX)∊Y)/X` `X←A; Y←I1`
Removing columns Y from array X

397 `(¯1↑(' '≠X)/⍳ρX)ρX` `X←C1`
Removing trailing blanks

398 `(1-⍳¯1↑ρX)⌽X` `X←A2`
Aligning columns of matrix X to diagonals

399 `(¯1+⍳¯1↑⍴X)⌽X` *X←A2*
Aligning diagonals of matrix X to columns

400 `0∊1↑0⍴X` *X←A*
Test if numeric

401 `(¯1+X⍳' ')↑X` *X←C1*
First word in X

402 `(¯2↑1 1,⍴X)⍴X` *X←A; 2≥⍴⍴X*
Reshaping non-empty lower-rank array X into a matrix

403 `1↑⍞,X` *X←C0*
Giving a character default value for input

404 `(X×Y)⍴(-Y)↑1` *X←I0; Y←I0*
Ending points for X fields of width Y

405 `(X×Y)⍴Y↑1` *X←I0; Y←I0*
Starting points for X fields of width Y

406 `X+(-⍴X)↑Y` *X←D; Y←D0*
Adding scalar Y to last element of X

407 `Y↑X⍴1` *X←I0; Y←I0*
Vector of length Y with X ones on the left, the rest zeroes

408 `1 80⍴80↑X` *X←A1*
Forming first row of a matrix to be expanded

409 `1↑0⍴X` *X←A*
Zero or space depending on the type of X (fill element)

410 `¯1↑⍴X` *X←A2*
Number of columns in matrix X

411 `1↑⍴X` *X←A2*
Number of rows in matrix X

412 `(-Y)↑X` *Y←I0; X←C1*
Transferring text X to right edge of field of width Y

DROP ↓

413 `(1↓A)-¯1↓A←(A,1)/⍳1+⍴A←1,(1↓X)≠¯1↓X` *X←A1*
Lengths of subvectors with equal elements

414 `G-¯1↓0,G←(~⎕IO)+(((1↓X)≠¯1↓X),1)/⍳⍴X` *X←A1; G←I1*
Field lengths of vector X; G ↔ ending indices

415 `(A>0)/A←(1↓A)-1+¯1↓A←(~A)/⍳⍴A←0,X,0` *X←B1*
Lengths of groups of ones in X

```
416  ((A⍳1)-⎕IO)↓(⎕IO-(⌽A←~X∊Y)⍳1)↓X                   X←A1; Y←A
     Removing elements Y from beginning and end of vector X

417  ((1↓X)≠¯1↓X),1                                              X←A1
     Ending points of equal groups

418  1,(1↓X)≠¯1↓X                                                X←A1
     Starting points of equal groups

419  (1↓X)÷¯1↓X                                                  X←D1
     Pairwise ratios of successive elements of vector X

420  (1↓X)-¯1↓X                                                  X←D1
     Pairwise differences of successive elements of vector X

421  (Y∧X)∨(Y∨X)\A>¯1↓0,A←(Y∨X)/Y                         X←B1; Y←B1
     First ones of subvectors of Y indicated by X (<\)

422  (¯1⌽1↓(X≠¯1⌽X),1)/X                                         X←A1
     Removing duplicates from an ordered vector

423  A-¯1↓0,A←(1⌽X)/⍳⍴X                                          X←B1
     Changing starting indicators of subvectors to lengths

424  (A∨1↓1⌽1,A←' '≠X)/X                                         X←C1
     Removing multiple blanks

425  (⍴X)⍴A ∆ A[(A=1⍴Y)/⍳⍴A←,X]←1↓Y              X←A; Y←A1; 2 ↔ ⍴Y
     Replacing all values Y[1] by Y[2] in X

426  (A∨¯1↓1,A←X≠Y)/X                                      X←A1; Y←A0
     Removing duplicate Y's from vector X

427  (⎕IO-(~⌽X∊Y)⍳1)↓X                                      X←A1; Y←A
     Removing elements Y from end of vector X

428  (1-(⌽' '≠X)⍳1)↓X                                           X←C1
     Removing trailing blanks

429  0 ¯1↓(-⍳⍴X)⌽((2⍴⍴X)⍴0),X                                    X←D1
     Diagonal matrix with elements of X

430  ¯1↓X×⌽¯1+⍳⍴X                                                X←D1
     Derivate of polynomial X

431  (¯1↓X≠1⌽X),1                                                X←A1
     Test if an element differs from the next one

432  1,1↓X≠¯1⌽X                                                  X←A1
     Test if an element differs from the previous one

433  1⌽¯1↓Y,X                                              X←A1; Y←A0
     Replacing last element of X with Y
```

```
434  ¯1⌽1↓X,Y                                      X←A1; Y←A0
     Replacing first element of X with Y

435  (((~X∊Y)⍳1)-⎕IO)↓X                             X←A1; Y←A
     Removing elements Y from beginning of vector X

436  X-(-Y=⍳⍴⍴X)↓0,[Y]X                             X←D; Y←I0
     Differences of successive elements of X along direction Y

437  (¯1+(X='0')⍳0)↓X                                     X←A1
     Removing leading zeroes

438  Y+(Y↓X)⍳1                                     G←I0; X←B1
     Index of first one after index Y in X

439  X>1↓X,0                                              X←B1
     Last ones in groups of ones

440  X>¯1↓0,X                                             X←B1
     First ones in groups of ones

441  1↓,',',X                                             X←C2
     List of names in X (one per row)

442  X-¯1↓0,X                                             X←D1
     Restoring argument of cumulative sum (inverse of +\)

443  (0,Y)↓X                                         X←A2;Y←I0
     Drop of Y first columns from matrix X

444  (Y,0)↓X                                        X←A2; Y←I0
     Drop of Y first rows from matrix X

445  1↓⍴X                                                 X←A2
     Number of columns in matrix X

446  ¯1↓⍴X                                                X←A2
     Number of rows in matrix X

447  (Y×G)↓X                                  X←A; Y←I1; G←B1
     Conditional drop of Y elements from array X

448  (-Y)↓X                                         X←A1; Y←B0
     Conditional drop of last element of X

---
     CEILING / MAXIMUM  ⌈

449  Y[1]⌈Y[2]⌊X                                     X←D; Y←D1
     Limiting X between Y[1] and Y[2], inclusive

---
     FLOOR / MINIMUM  ⌊
```

```
450  ⌊|10⍟|1-3×÷3
     Arithmetic precision of the system (in decimals)

451  X+(G××Y-X)×(⍳1+|⌊(Y-X)÷G)-⎕IO             X←D0; Y←D0; G←D0
     Arithmetic progression from X to Y with step G

452  1+(X<0)+⌊10⍟|X+0=X                                      X←I
     Number of digitpositions in integers in X

453  ⌊X+1≤2|X                                                X←D
     Rounding to nearest even integer

454  ⌊X+.5×.5≠2|X                                            X←D
     Rounding, to nearest even integer for .5 = 1||X

455  1+⌊10⍟(X=0)+X×1 ¯10[1+X<0]                              X←I
     Number of digit positions in integers in X

456  1+⌊10⍟X+0=X                                             X←I
     Number of digits in positive integers in X

457  X=⌊X                                                    X←D
     Test if integer

458  (X,G)[(1+⍴X)⌊Y]                             X←A1; Y←I; G←A0
     Limiting indices and giving a default value G

459  ⌊X÷1000                                                 X←I
     First part of numeric code ABBB

460  (10*-X)×⌊0.5+Y×10*X                               X←I; Y←D
     Rounding to X decimals

461  0.01×⌊0.5+100×X                                         X←D
     Rounding to nearest hundredth

462  ⌊0.5+X                                                  X←D
     Rounding to nearest integer

---
     MAGNITUDE / RESIDUE  |

463  (0=400|X)∨(0≠100|X)∧0=4|X                               X←I
     Test if X is a leap year

464  '_',[1]('|',X,'|'),[1]'¯'                              X←C2
     Framing

465  1||X                                                    X←D
     Magnitude of fractional part

466  (0≠Y|⍳⍴X)/X                                       X←A1; Y←I0
     Removing every Y:th element of X
```

```
467  (0=Y|⍳⍴X)/X                                          X←A1; Y←I0
     Taking every Y:th element of X

468  (0=A|X)/A←⍳X                                              X←I0
     Divisors of X

469  (2|⍳⍴X)/X                                                 X←A1
     Removing every second element of X

470  (0=Y|X)/X                                    X←D1; Y←D0 or D1
     Elements of X divisible by Y

471  ⎕IO+(⍴X)|Y+(Y⌽X)⍳G                           X←A1; Y←I0; G←A
     Index of first occurrence of G in X (circularly) after Y

472  (1+⍴Y)|Y⍳X                                          Y←A1; X←A
     Changing index of an unfound element to zero

473  ~2|X                                                        X←I
     Test if even

474  X×Y≤|X                                               X←D; Y←D
     Rounding to zero values of X close to zero

475  (×X)×Y+|X                                            X←D; Y←D
     Increasing absolute value without change of sign

476  (×X)|X                                                      X←D
     Fractional part with sign

477  X×|X                                                        X←D
     Square of elements of X without change of sign

478  1|X                                                         X←D
     Fractional part

479  1000|X                                                      X←I
     Last part of numeric code ABBB

---
     EXPAND  \

480  A\(A←~X∊Y)/X                                         X←A0; Y←A1
     Replacing elements of X in set Y by blanks / zeroes

481  A\(A←X∊Y)/X                                           X←A1; Y←A
     Replacing elements of X not in set Y by blanks / zeroes

482  A ∆ A[(~G)/⍳⍴G]←Y  ∆  A←G\X                 X←A1; Y←A1; G←B1
     Merging X and Y under control of G (mesh)

483  Y\Y/X                                                X←A; Y←B1
     Replacing elements of X not fulfilling Y by blanks / zeroes
```

```
484  ⌽α\⌽X                                                X←A
     Scan from end with α

485  (~(⍳(⍴Y)+1⍴⍴X)∊Y+⍳⍴Y)⍀X                      X←A2; Y←I1
     Adding an empty row into X after rows Y

486  0∊0\0⍴X                                             X←A1
     Test if numeric

487  ((Y+1)≠⍳1+1⍴⍴X)⍀X                             X←A2; Y←I0
     Adding an empty row into X after row Y

488  X,[⎕IO-.1](' '≠X)\'¯'                               X←C1
     Underlining words

489  (⍴Y)⍴(,Y)\X                                   X←A1; Y←B2
     Using boolean matrix Y in expanding X

490  ((2×⍴X)⍴1 0)\X                                      X←C1
     Spacing out text
```

———

COMPRESS /

```
491  (X/Y)≥A/1⌽A←(Y∨X)/X                           X←B1; Y←B1
     Or-reduction (∨/) of subvectors of Y indicated by X

492  (X/Y)∧A/1⌽A←(Y≤X)/X                           X←B1; Y←B1
     And-reduction (∧/) of subvectors of Y indicated by X

493  (G/X),(~G)/Y                           X←C1; Y←C1; G←B0
     Choosing a string according to boolean value G

494  (A∨1⌽A←X≠' ')/X                                     X←C1
     (Cyclic) compression of successive blanks

495  (X∊Y)/⍳⍴X                                      X←A1; Y←A
     Indices of all occurrences of elements of Y in X

496  (~X∊' .,:;?''')/X                                   X←A1
     Removing punctuation characters

497  Y,(~X∊Y)/X                                    X←A1; Y←A1
     Union of sets, ∪

498  (~X∊Y)/X                                       X←A1; Y←A
     Elements of X not in Y (difference of sets)

499  (X[;1]∊Y)⌿X                                   X←A2; Y←A1
     Rows of non-empty matrix X starting with a character in Y

500  (X∊Y)/X                                        X←A1; Y←A
     Intersection of sets, ∩
```

501 ((X⍳X)=⍳⍴X)/X   X←A1
Removing duplicate elements (nub)

502 ((⍴X)*Y≠⍳⍴⍴X)⍴ α/[Y]X   Y←I0; X←A
Reduction in dimension Y, rank unchanged

503 (Y=X)/⍳⍴X   X←A1; Y←A0
Indices of all occurrences of Y in X

504 Y[X/⍳⍴Y]←G   Y←A1 X←B1 G←A0
Replacing elements of Y satisfying X with G

505 A/⍳9999 Δ A[X]←1 Δ A←9999⍴0   X←I1
Removing duplicates from positive integers

506 X/⍳⍴X   X←B1
Indices of ones in logical vector X

507 (,X,[1.5]1)/,X,[1.5]~X   X←B1
Vector to expand a new element after each one in X

508 ((~X)/'IN'),'CORRECT'   X←B0
Conditional in text

509 (X≠Y)/X   X←A1; Y←A0
Removing elements Y from vector X

510 (' '≠X)/X   X←A1
Removing blanks

511 α/,X   X←D
Reduction with α without respect to shape

512 Y/X   X←A; Y←B1
Selecting elements of X satisfying condition Y

513 0⌿X   X←A2
Empty matrix

TRANSPOSE ⍉

514 ⍉F⍉X   X←A1; Y←I0
Applying to columns function F defined on rows

515 ((⌽A)×1,Y)⍴2 1 3⍉(1⌽Y,A←(⍴X)÷1,Y)⍴X   X←A2; G←I0
"Transpose" of matrix X with column fields of width Y

516 X×⍉(⌽⍴X)⍴Y   X←D2; Y←D1
Multiplication of each column of matrix X by vector Y

517 ⍉(⌽⍴Y)⍴X   X←A1; Y←A2
Matrix with shape of Y and X as its columns

```
518  (Y⌽1 2)⍉X                                          X←A2; Y←B0
     Transpose matrix X on condition Y

519  ~0∊X=-⍉X                                                 X←D2
     Test for antisymmetricity of square matrix X

520  ~0∊X=⍉X                                                  X←A2
     Test for symmetricity of square matrix X

521  ⍉(X,⍴Y)⍴Y                                          X←I0; Y←D1
     Matrix with X columns Y

522  1 1⍉X[Y[1;];Y[2;]]                                 X←A2; Y←I2
     Retrieving scattered elements Y from matrix X

523  X[Y]⍉G                                             X←I1; Y←I1
     Successive transposes of G (X after Y: X⍉Y⍉G)

524  40 120⍴2 1 3⍉10 40 12⍴X                                  X←C2
     Dividing a 400×12 character matrix into one page

525  (1*⍴X)⍉X                                                  X←A
     Main diagonal of array X

526  1 1⍉X                                                    X←A2
     Diagonal elements of matrix X

527  1 3 2⍉X                                                  X←A3
     Transpose of planes of a rank three array

---
```

REVERSE / ROTATE ⌽ ⊖

```
528  ((1⌽X)×¯1⌽Y)-(¯1⌽X)×1⌽Y                              X←D; Y←D
     Vector (cross) product of vectors

529  1⌽(⍳⍴X)∊Y                                          X←A1; Y←I1
     Ending points for X in indices pointed by Y

530  (¯1 1[2×⎕IO]+⍴X)-(⌽X)⍳Y                            X←A1; Y←A0
     Index of last element of Y in X

531  (1+⍴X)-(⌽X)⍳Y                                       X←A1; Y←A
     Index of last occurrence

532  (⌽X)⍳Y                                              X←A1; Y←A
     Index of last occurrence, counted from the rear

533  ,⌽[⎕IO+Y](1,⍴X)⍴X                                  X←A1; Y←B0
     Reverse vector X on condition Y

534  (⌽1,⍴X)⍴X                                                X←A1
     Reshaping vector X into a one-column matrix
```

535 (⌽1, ...) ...←EXPRESSION
Avoiding parentheses with help of reversal

536 1⌽X X←A
Rightmost neighboring elements cyclically

537 ¯1⌽X X←A
Leftmost neighboring elements cyclically

MEMBER OF ∊

538 ~(⍳(⍴Y)+⍴X)∊Y+⍳⍴Y X←A1; Y←I1
Expansion vector with zero after indices Y

539 (~(⍳Y)∊X) X←I; Y←I0
Boolean vector of length Y with zeroes in locations X

540 (⍳⍴X)∊Y X←A1; Y←I1
Starting points for X in indices pointed by Y

541 (X∊Y)×Y⍳X X←A; Y←A1
Changing index of an unfound element to zero (not effective)

542 (Y←⎕)∊⍳X X←A
Check for input in range 1..X

543 (⍳Y)∊X X←I; Y←I0
Boolean vector of length Y with ones in locations X

544 ~0∊X=Y X←A; Y←A
Test if arrays are identical

545 Y×~Y∊X Y←D; X←D
Zeroing elements of Y depending on their values

546 1∊⍴,X X←A
Test if single or scalar

547 1∊⍴⍴X X←A
Test if vector

548 0∊⍴X X←A
Test if X is an empty array

INDEX GENERATOR / INDEX OF ⍳

549 (G⍳X)<G⍳Y X←A; Y←A
Alphabetical comparison with alphabets G

550 X⍳⍳⍴X X←I1
Inverting a permutation

```
551  (Y≠X)⍳1                                           X←A1; Y←A1
     Index of first differing element in vectors X and Y

552  (⎕IO+⍴Y)=Y⍳X                                       X←A; Y←A1
     Which elements of X are not in set Y (difference of sets)

553  A Δ A[X]←A Δ A←⍳⍴X                                       X←I1
     Inverting a permutation

554  G[Y⍳X;]                                     X←D; Y←D1; G←C2
     Changing numeric code X into corresponding name in Y

555  ⍳⍴⍴X                                                       X←A
     All axes of array X

556  ⍳⍴X                                                       X←A1
     All indices of vector X

557  X+G×(⍳Y)-⎕IO                               X←D0; Y←D0; G←D0
     Arithmetic progression of Y numbers from X with step G

558  (X-⎕IO)+⍳1+Y-X                                    X←I0; Y←I0
     Consecutive integers from X to Y (arithmetic progression)

559  X⍳1                                                       X←B1
     Index of first satisfied condition in X

560  ⍳1
     Index origin (⎕IO) as a vector

561  ⎕AV⍳X                                                      X←A
     Converting characters into numeric codes

562  X⍳Y                                                 X←A1; Y←A
     Index of key Y in key vector X

563  ⍳0
     Empty numeric vector

---
     LOGICAL ~ ∨ ∧ ⍱ ⍲

564  (Y[1]<X)∧X<Y[2]                                    X←D; Y←D1
     Test if X is within range ( Y[1],Y[2] )

565  (Y[1]≤X)∧(X≤Y[2])                            X←D; Y←D1; 2=⍴Y
     Test if X is within range [ Y[1],Y[2] ]

566  0∧X                                                        X←B
     Zeroing all boolean values

567  (X×G)+Y×~G                                    X←D; Y←D; G←B
     Selection between X and Y depending on condition G
```

568 `(~⎕IO)+X` X←I
Changing an index origin dependent result to be as ⎕IO=1

569 `Y*~X` Y←D; X←B
Conditional change of elements of Y to one according to X

COMPARISON < ≤ = ≥ > ≠

570 `X≤Y` X←B; Y←B
X implies Y

571 `X>Y` X←B; Y←B
X but not Y

572 `(0≠X)×Y÷X+0=X` X←D; Y←D
Avoiding division by zero error (gets value zero)

573 `X≠Y` X←B; Y←B
Exclusive or

574 `X+Y×X=0` X←D; Y←D
Converting zeroes to Y

575 `Y=X` X←I; Y←I
Kronecker delta of X and Y (element of identity matrix)

RAVEL / CATENATE ,

576 `,(((ρX),Y)ρG),X` X←A1; Y←I0; G←A0
Catenating Y elements G before every element of X

577 `,X,((ρX),Y)ρG` X←A1; Y←I0; G←A
Catenating Y elements G after every element of X

578 `,Y,[⎕IO+.5]X` X←A1; Y←A1
Merging vectors X and Y alternately

579 `X,⎕TC[2],Y` X←A1; Y←A1
Separating variable length lines

580 `,X,[1.1]' '` X←C1
Spacing out text

581 `,X,[1.1]Y` X←A1; Y←A0
Inserting Y after each element of X

582 `(X,X)ρ1,Xρ0` X←I0
X×X identity matrix

583 `X,[.5+ρρX]-X` X←D
Array and its negative ('plus minus')

584 X←(ρX)ρA Δ A[G]←Y Δ A←,X X←A; Y←A; G←I
Temporary ravel of X for indexing with G

585 X←AρX Δ X[G]←Y Δ X←,X Δ A←ρX X←A; Y←A; G←I
Temporary ravel of X for indexing with G

586 X,[□IO-.1]'¯' X←C1
Underlining a string

587 X[;,1] X←A2
First column as a matrix

588 X,[.1]Y X←A1; Y←A1
Forming a two-row matrix

589 X,[1.1]Y X←A1; Y←A1
Forming a two-column matrix

590 ((((ρρX)-ρρY)ρ1),ρY)ρY X←A; Y←A
Increasing rank of Y to rank of X

591 ((0.5×ρX),2)ρX X←A1
Reshaping vector X into a two-column matrix

592 (ρX)ρ ... ,X X←A; ...←EXPRESSION
Handling array X temporarily as a vector

593 (Y,ρX)ρX X←A1; Y←I0
Forming a Y-row matrix with all rows alike (X)

594 ((ρX),1)ρX X←A1
Reshaping vector X into a one-column matrix

595 (1,ρX)ρX X←A1
Reshaping vector X into a one-row matrix

596 ρ,X X←A
Number of elements (also of a scalar)

597 Y,0ρX X←A; Y←A1
Joining sentences

598 X←0 2 1 2 5 8 0 4 5,□ X←D
Entering from terminal data exceeding input (printing) width

AXIS / INDEXING [

599 (ρX)[ρρX] X←A
Number of columns in array X

600 (ρX)[2] X←A2
Number of columns in matrix X

601 `(ρX)[1]` X←A2
Number of rows in matrix X

602 `Y[2+×X]` X←D; Y←A1
Choosing according to signum

603 `Y×1 ¯1[1+X]` Y←D; X←B
Conditional elementwise change of sign

604 `X[2×⎕IO]` X←A1
Selection depending on index origin

605 `' *'[⎕IO+X]` X←B
Indexing with boolean value X (plotting a curve)

606 `X[⎕IO+Y]` X←A1; Y←I
Indexing independent of index origin

607 `X[;1]` X←A2
First column as a vector

608 `X[]←0` X←D1
Zeroing a vector (without change of size)

609 `X[1]` X←A1
Selection depending on index origin

SHAPE / RESHAPE ρ

610 `(Y×ρX)ρX` X←A1; Y←I0
Duplicating vector X Y times

611 `X×(ρX)ρY` X←D; Y←D1
Multiplication of each row of X by vector Y

612 `ρρX` X←A
Rank of array X

613 `1ρρX` X←A2
Number of rows in matrix X

614 `(ρY)ρX` X←A1; Y←A
Array with shape of Y and X as its rows

615 `1ρX` X←A
Corner element of a (non-empty) array

616 `''ρX` X←A
Reshaping X into a scalar

617 `0ρX←` X←A
Output of an empty line

618 `0 80⍴0`
Forming an initially empty array to be expanded

\-\-\-

ARITHMETIC + − × ÷

619 `1+÷2+÷3+÷4+÷5+÷6 ...`
Continued fraction

620 `Y×÷X` — *X←D; Y←D*
Force 0÷0 into DOMAIN ERROR in division

621 `1 ¯1×X` — *X←D0*
Number and its negative ('plus minus')

622 `Y×X` — *X←D; Y←B*
Selecting elements satisfying condition Y, zeroing others

623 `X×¯1*Y` — *X←D; Y←B; ⍴X ←→ ⍴Y*
Conditional elementwise change of sign

624 `0×X` — *X←D*
Zero array of shape and size of X

625 `-⎕IO-X` — *X←I*
Changing an index origin dependent result to be as ⎕IO=0

626 `(⎕IO-1)+X` — *X←I*
Changing an index origin dependent argument to act as ⎕IO=1

627 `+X←` — *X←D*
Output of assigned numeric value

628 `⎕IO+X` — *X←I*
Changing an index origin dependent argument to act as ⎕IO=0

\-\-\-

MISCELLANEOUS

629 `*`
Syntax error to stop execution

630 `⎕←X←` — *X←A*
Output of assigned value

631 `⎕LX←⍞`
Setting latent expression

SYNONYM LIST

Adding X to each column of Y

```
193  1 1 2⍉X∘.+Y
192  1 2 1⍉Y∘.+X
```

Adding X to each row of Y

```
190  2 1 2⍉X∘.+Y
189  1 2 2⍉Y∘.+X
```

Arithmetic average (mean value)

```
365  (+/X)÷⍴X
332  (+/,X)÷1⌈⍴,X
```

Ascending cardinal numbers (ranking)

```
  1  ⌊.5×(⍋⍋X)+⌽⍋⍋⌽X
 13  ⍋⍋X
```

Centering text

```
323  Y↑((⌊-/.5×Y,⍴X)⍴' '),X
393  (-⌊.5×Y+⍴X)↑X
269  (-⌊0.5×+/∧\' '=⌽X)⌽X
248  (A-⌊0.5×(A←+/∧\⌽A)++/∧\A←' '=⌽X)⌽X
```

Changing index of an unfound element to zero

```
472  (1+⍴Y)|Y⍳X
541  (X∊Y)×Y⍳X
```

Conditional elementwise change of sign

```
603  Y×1 ¯1[1+X]
623  X×¯1*Y
```

Diagonal elements of matrix X

```
526  1 1⍉X
525  (1*⍴X)⍉X
```

Drop of Y first rows from matrix X

```
444  (Y,0)↓X
388  (2↑Y)↓X
```

Evaluating a two-row determinant

```
314  -/×/0 1⊖X
313  -/×⌿0 1⌽X
```

Expansion vector with zero after indices Y

```
 27  (⍴X)≥⍋(⍳⍴X),Y
538  ~(⍳(⍴Y)+⍴X)∊Y+⍳⍴Y
```

First ones in groups of ones

```
381  X>((-⍴⍴X)↑¯1)↓0,X
440  X>¯1↓0,X
```

Giving a default value for input

```
 97  1⍴(⍎⍞,',⍳0'),X
403  1↑⍞,X
```

Index of (first) maximum element of X

```
 23  ¯1↑⍋X
330  X⍳⌈/X
```

Index of (first) minimum element of X

```
 22  1↑⍋X
336  X⍳⌊/X
```

Indices of ones in logical vector X

```
 41  (+/X)↑⍒X
280  (+\X)⍳⍳+/X
506  X/⍳⍴X
```

Inverting a permutation

```
 37  ⍋X
553  A Δ A[X]←A Δ A←⍳⍴X
550  X⍳⍳⍴X
```

Locations of texts between and including quotes

```
287  A∨¯1↓0,A←2|+\X=''''
293  A∨¯1↓0,A←≠\X=''''
```

Locations of texts between quotes

```
288  A∧¯1↓0,A←2|+\X=''''
294  A∧¯1↓0,A←≠\X=''''
```

Merging (mesh)

```
482  A Δ A[(~G)/⍳⍴G]←Y  Δ  A←G\X
 12  (X,Y)[⍋⍋G]
 16  (Y,X)[⍋⍒G]
 31  A Δ A[⍋G]←A←Y,X
 11  (Y,X,Z,...)[⍋⍋G]
```

Number and its negative ('plus minus')

```
621  1 ¯1×X
583  X,[.5+⍴⍴X]-X
204  X∘.×1 ¯1
```

Number of columns in matrix X

```
600  (⍴X)[2]
410  ¯1↑⍴X
445  1↓⍴X
599  (⍴X)[⍴⍴X]
```

Number of elements (also of a scalar)

```
366  ×/⍴X
596  ⍴,X
```

Number of occurrences of scalar X in array Y

```
236  X+.=,Y
362  +/X=,Y
```

Number of rows in matrix X

```
601  (⍴X)[1]
411  1↑⍴X
```

```
613  1ρρX
446  ¯1↓ρX
```

Occurrence(s) of string X in string Y

```
178  (∧⌿(¯1+ιρX)⌽X∘.=Y)ι1
147  (Y[A∘.+¯1+ιρX]∧.=X)/A←(A=1↑X)/ιρA←(1-ρX)↓Y
219  ((-A)↓X∧.=(A,1+ρY)ρY)/ι(ρY)+1-A←ρX
```

Output of assigned value

```
630  ⎕←X←
627  +X←
```

Removing duplicate elements

```
501  ((XιX)=ιρX)/X
154  (1 1⍉<\X∘.=X)/X
505  A/ι9999 Δ A[X]←1 Δ A←9999ρ0
422  (¯1⌽1↓(X≠¯1⌽X),1)/X
```

Removing duplicate rows

```
 70  ((AιA)=ιρA←2⊥X∧.=⍉X)⌿X
208  (1 1⍉<\X∧.=⍉X)⌿X
327  (¯1⌽1↓(∨/X≠¯1⊖X),1)⌿X
206  ((AιA)=ιρA←⎕IO++⌿∧⍀X∨.≠⍉X)⌿X
```

Removing leading blanks

```
267  (+/∧\' '=X)↓X
299  (∨\' '≠X)/X
```

Removing trailing blanks

```
298  (⌽∨\⌽' '≠X)/X
 73  (1-(' '=X)⊥1)↓X
397  (¯1↑(' '≠X)/ιρX)ρX
428  (1-(⌽' '≠X)ι1)↓X
266  (-+/∧\' '=⌽X)↓X
```

Removing trailing blank columns

```
272  (⌽∨\⌽∨⌿' '≠X)/X
209  (⌽∨\⌽' '∨.≠X)/X
```

Replicate Y[i] X[i] times (for all i)

```
250  Y[+\(ι+/X)∊¯1↓1++\0,X]
251  ((X≠0)/Y)[+\¯1⌽(ι+/X)∊+\X]
253  Y[⎕IO++\(ι+/X)∊⎕IO++\X]
```

Representations of current date

```
107  A Δ A[(' '=A)/ιρA]←'/' Δ A←⍕100|1⌽3↑⎕TS
 65  A Δ A[5 8]←'-' Δ A←⍕1000⊥3↑⎕TS
104  A Δ A[(' '=A)/ιρA]←'.' Δ A←⍕⌽3↑⎕TS
 72  100⊥100|3↑⎕TS
```

Representations of current time

```
 64  A Δ A[3 6]←':' Δ A←⍕1000⊥3↑3↓⎕TS
105  (,(0 1↓3 0⍕100+3 1ρ12 0 0|3↑3↓⎕TS),':: '),'AP'[1+12≤⎕TS[4]],'M'
```

Reshaping non-empty lower-rank array X into a matrix

```
392  (1⌈¯2↑ρX)ρX
402  (¯2↑1 1,ρX)ρX
```

Reshaping vector X into a one-column matrix

```
203  X∘.+,0
534  (⌽1,ρX)ρX
594  ((ρX),1)ρX
```

Selection depending on index origin

```
604  X[2×⎕IO]
609  X[1]
```

Sorting rows of matrix X into ascending order

```
  9  (ρX)ρ(,X)[A[⍋(,⍉(⌽ρX)ρι1↑ρX)[A←⍋,X]]]
  8  (ρX)ρ(,X)[⎕IO+A[⍋⌊A÷¯1↑ρX]]  Δ  A←(⍋,X)-⎕IO
```

Spacing out text

```
580  ,X,[1.1]' '
490  ((2×ρX)ρ1 0)\X
```

Temporary ravel of X for indexing with G

```
585  X←A⍴X Δ X[G]←Y Δ X←,X Δ A←⍴X
584  X←(⍴X)⍴A Δ A[G]←Y Δ A←,X
```

Test for symmetricity of matrix X

```
 86  ⍎⍎'1','↑↓'[⎕IO+∧/(⍴X)=⌽⍴X],'''0~0∊X=⍉X'''
520  ~0∊X=⍉X
```

Test if all elements of vector X are equal

```
358  ∧/X=X[1]
343  ∧/X=1⌽X
340  0=(⍴X)|+/X
159  ∨/∧/0 1∘.=X
324  ∧/X=1↑X
315  (∧/X)∨~∨/X
311  (⌈/X)=⌊/X
317  ∧/X÷∨/X
352  ≠/0 1∊X
316  (∧/X)=∨/X
347  ⍲/0 1∊X
215  X∧.=∧/X
214  X∧.=∨/X
341  ∧/X/1⌽X
354  ∧/⎕IO=X⍳X
349  ~∧/X∊~X
```

Test if arrays of equal shape are identical

```
357  ∧/,X=Y
544  ~0∊X=Y
```

Test if numeric

```
400  0∊1↑0⍴X
486  0∊0\0⍴X
```

Test if X is a permutation vector

```
 20  X[⍋X]∧.=⍳⍴X
158  ∧/1=+/X∘.=⍳⍴X
346  ∧/(⍳⍴X)∊X
```

Value of polynomial with coefficients Y at point X

```
227  (X*¯1+⍳⍴Y)+.×⌽Y
 83  X⊥Y
 69  (X∘.+,0)⊥Y
```

Value of Taylor series with coefficients Y at point X

```
137  +/Y×(X*A)÷!A←¯1+⍳⍴Y
281  +/Y××\1,X÷⍳¯1+⍴Y
```

Vectors as matrices in catenation

```
273  X,[⎕IO-(⍲\2⍴∧/2>(⍴⍴X),⍴⍴Y)/.5 0]Y
274  X,[⎕IO+(⍲\2⍴∧/2>(⍴⍴X),⍴⍴Y)/.5 1]Y
```

X×X identity matrix

```
197  (⍳X)∘.=⍳X
582  (X,X)⍴1,X⍴0
```

PERMUTED INDEX

First part of numeric code	ABBB	459
Last part of numeric code	ABBB	479
Decoding numeric code	ABBCCC into a matrix	63
Increasing	absolute value without change of sign	475
Choosing a string	according to boolean value G	493
Sums	according to codes G	152
Sorting indices X	according to data Y	32
Case structure	according to key vector G	117
Grade up	according to key Y	30
Sorting rows of X	according to key Y (alphabetizing)	18
Choosing	according to signum	602
Using a variable named	according to X	88
Sorting Y	according to X	36
Conditional change of elements of Y to one	according to X	569
Sorting a matrix	according to Y:th column	33
Barchart of integer values (	across the page)	171
Barchart of X with height Y (	across the page)	170
Barchart of two integer series (	across the page)	68
Changing an index origin dependent argument to	act as □IO=1	626
Changing an index origin dependent argument to	act as □IO=0	628
	Adding a new dimension after dimension G Y-fold	10
	Adding an empty row into X after rows Y	485
	Adding an empty row into X after row Y	487
	Adding scalar Y to last element of X	406
	Adding X to each column of Y	192
	Adding X to each column of Y	193
	Adding X to each row of Y	189
	Adding X to each row of Y	190
Adding a new dimension	after dimension G Y-fold	10
Inserting Y	after each element of X	581
Vector to expand a new element	after each one in X	507
Catenating Y elements G	after every element of X	577
Turn off all ones	after first one	307
Turn on all zeroes	after first one	304
Turn off all ones	after first zero	306
Turn on all zeroes	after first zero	308
Insert vector for X[I] zeroes	after I:th subvector	265
Inserting X into Y	after index G	382
Index of first one	after index Y in X	438
Inserting Y *'s into vector X	after indices G	26
Expansion vector with zero	after indices Y	27
Expansion vector with zero	after indices Y	538
Catenating G elements H	after indices Y in vector X	28
Inserting vector X into matrix Y	after row G	375
Adding an empty row into X	after row Y	487
Adding an empty row into X	after rows Y	485
Index of first occurrence of G in X (circularly)	after Y	471
Successive transposes of G (X	after Y: X⍉Y⍉G)	523
Open a gap of X[I]	after Y[G[I]] (for all I)	264
	Aligning columns of matrix X to diagonals	398
	Aligning diagonals of matrix X to columns	399
Forming a Y-row matrix with all rows	alike (X)	593
	All, both	360
	All axes of array X	555
	All binary representations up to X (truth table)	52
	All binary representations with X bits (truth table)	60
Moving	all blanks to end of each row	150
Moving	all blanks to end of text	42
Zeroing	all boolean values	566
Ascending cardinal numbers (ranking,	all different)	13
Descending cardinal numbers (ranking,	all different)	17
Test if	all elements of vector X are equal	159
Test if	all elements of vector X are equal	214
Test if	all elements of vector X are equal	215
Test if	all elements of vector X are equal	311
Test if	all elements of vector X are equal	315

Test if all elements of vector X are equal 316
Test if all elements of vector X are equal 317
Test if all elements of vector X are equal 324
Test if all elements of vector X are equal 340
Test if all elements of vector X are equal 341
Test if all elements of vector X are equal 343
Test if all elements of vector X are equal 347
Test if all elements of vector X are equal 349
Test if all elements of vector X are equal 352
Test if all elements of vector X are equal 354
Test if all elements of vector X are equal 358
Replicate Y[i] X[i] times (for all i) 250
Replicate Y[i] X[i] times (for all i) 251
Replicate Y[i] X[i] times (for all i) 253
Open a gap of X[i] after Y[G[i]] (for all i) 264
Open a gap of X[i] before Y[G[i]] (for all i) 271
All indices of vector X 556
Matrix of all indices of X 51
Indices of all occurrences of elements of Y in X 495
Indices of all occurrences of Y in X 503
Turn off all ones after first one 307
Turn off all ones after first zero 306
All pairs of elements of ⍳X and ⍳Y 58
All primes up to X 175
Forming a Y-row matrix with all rows alike (X) 593
Matrix for choosing all subsets of X (truth table) 59
Replacing all values Y[1] by Y[2] in X 425
Turn on all zeroes after first one 304
Turn on all zeroes after first zero 308
Test if X is in ascending order along direction Y 268
Test if X is in descending order along direction Y 270
Average (mean value) of elements of X along direction Y 364
Differences of successive elements of X along direction Y 436
Alphabetical comparison with alphabets G 549
Sorting rows of X according to key Y (alphabetizing) 18
Alphabetizing X; equal alphabets in same column of Y 395
Alphabetical comparison with alphabets G 549
Alphabetizing X; equal alphabets in same column of Y 395
Transformation of alphanumeric string into numeric 78
Arithmetic average (mean value), also for an empty array 332
Positive maximum, at least zero (also for empty X) 334
Number of elements (also of a scalar) 366
Number of elements (also of a scalar) 596
Number of rows in array X (also of a vector) 328
Merging vectors X and Y alternately 578
Alternating product 367
Sum of alternating reciprocal series Y+X 238
Alternating series (1 ¯1 2 ¯2 3 ¯3 ...) 301
Alternating sum 369
Contour levels Y at points with altitudes X 179
Representation of current date (American) 107
And-reduction (∧/) of subvectors of Y indicated by X 492
And-scan (∧\) over subvectors of Y indicated by X 290
Complementary angle (arccos sin X) 131
Rotation matrix for angle X (in radians) counter-clockwise 132
Annuity coefficient: X periods at interest rate Y % 186
Test for antisymmetricity of matrix X 85
Test for antisymmetricity of square matrix X 519
Any, anyone 356
Any, anyone 356
Test if X is a valid APL name 322
Applying to columns function F defined on rows 514
Complementary angle (arccos sin X) 131
Arctan Y+X 129
Area of triangle with side lengths in X (Heron's formula) 318
Restoring argument of cumulative sum (inverse of +\) 442

Changing an index origin dependent	argument to act as ⎕IO=1	626
Changing an index origin dependent	argument to act as ⎕IO=0	628
Executing costly monadic function F on repetitive	arguments	151
	Arithmetic average (mean value) of X weighted by Y	237
	Arithmetic average (mean value)	365
	Arithmetic average (mean value), also for an empty array	332
	Arithmetic precision of the system (in decimals)	450
Consecutive integers from X to Y (	arithmetic progression)	558
	Arithmetic progression from X to Y with step G	451
	Arithmetic progression of Y numbers from X with step G	557
Test if X is an empty	array	548
Corner element of a (non-empty)	array	615
Transpose of planes of a rank three	array	527
Conversion of indices to indices of raveled	array	81
Arithmetic average (mean value), also for an empty	array	332
	Array and its negative ('plus minus')	204
	Array and its negative ('plus minus')	583
Y-shaped	array of random numbers within (X[1],X[2]]	121
Zero	array of shape and size of X	624
Forming an initially empty	array to be expanded	618
	Array with shape of Y and X as its rows	614
Rank of	array X	612
Main diagonal of	array X	525
All axes of	array X	555
Removing columns Y from	array X	396
Number of columns in	array X	599
Indices of elements Y in	array X	55
Test if vector Y is a row of	array X	228
Comparing vector Y with rows of	array X	232
Indices of first blanks in rows of	array X	282
Conditional drop of Y elements from	array X	447
Number of rows in	array X (also of a vector)	328
Reshaping non-empty lower-rank	array X into a matrix	392
Reshaping non-empty lower-rank	array X into a matrix	402
Handling	array X temporarily as a vector	592
Centering character	array X with only right edge ragged	269
Centering character	array X with ragged edges	248
Replicating a dimension of rank three	array X Y-fold	200
Number of occurrences of scalar X in	array Y	236
Number of occurrences of scalar X in	array Y	362
Test if	arrays are identical	544
Test if	arrays of equal shape are identical	357
	Ascending cardinal numbers (ranking, shareable)	1
	Ascending cardinal numbers (ranking, all different)	13
Representation of current date (	ascending format)	104
Sorting X into	ascending order	35
Sorting rows of matrix X into	ascending order	8
Sorting rows of matrix X into	ascending order	9
Test if X is in	ascending order along direction Y	268
	Assign values of expressions in X to variables named in Y	98
Output of	assigned numeric value	627
Output of	assigned value	630
Arithmetic	average (mean value)	365
Arithmetic	average (mean value), also for an empty array	332
	Average (mean value) of columns of matrix X	326
	Average (mean value) of elements of X along direction Y	364
	Average (mean value) of rows of matrix X	325
Arithmetic	average (mean value) of X weighted by Y	237
Poisson distribution of states X with	average number Y	140
	Avoiding division by zero error (gets value zero)	572
	Avoiding parentheses with help of reversal	535
All	axes of array X	555
Histogram (distribution	barchart; down the page)	144
	Barchart of integer values (across the page)	171
	Barchart of integer values (down the page)	166
	Barchart of two integer series (across the page)	68

Barchart of X with height Y (across the page) 170
Representation of X in base Y 54
Catenating Y elements G before every element of X 576
Insert vector for X[i] elements before i:th element 279
Catenating G elements H before indices Y in vector X 29
Open a gap of X[i] before Y[G[i]] (for all i) 271
Indexing when rank is not known beforehand 100
Removing elements Y from beginning and end of vector X 416
Removing elements Y from beginning of vector X 435
Which class do elements of X belong to 181
Duplicating element of X belonging to Y,1↑X until next found 297
Predicted values of best linear fit (least squares) 125
Best linear fit of points (X,Y) (least squares) 128
Beta function 139
Locations of texts between and including quotes 287
Locations of texts between and including quotes 293
Places between pairs of ones 305
Locations of texts between quotes 288
Locations of texts between quotes 294
Selection between X and Y depending on condition G 567
Distribution of X into intervals between Y 145
Limiting X between Y[1] and Y[2], inclusive 449
Binary format of decimal number X 45
Extending a transitive binary relation 230
All binary representations up to X (truth table) 52
All binary representations with X bits (truth table) 60
Pascal's triangle of order X (binomial coefficients) 135
Binomial distribution of X trials with probability Y 138
All binary representations with X bits (truth table) 60
Index of first non-blank, counted from the rear 79
Removing duplicate blank columns 220
Removing leading blank columns 210
Removing trailing blank columns 209
Removing trailing blank columns 272
Removing blank columns 226
Blank rows 359
Removing duplicate blank rows 218
Removing leading blank rows 211
Removing trailing blank rows 205
Removing blank rows 225
Removing multiple blanks 424
Removing leading blanks 267
Removing leading blanks 299
Removing trailing blanks 73
Removing trailing blanks 266
Removing trailing blanks 298
Removing trailing blanks 397
Removing trailing blanks 428
Removing blanks 510
Removing leading and trailing blanks 259
(Cyclic) compression of successive blanks 494
Formatting with zero values replaced by blanks 108
Replacing elements of X in set Y by blanks / zeroes 480
Replacing elements of X not in set Y by blanks / zeroes 481
Replacing elements of X not fulfilling Y by blanks / zeroes 483
Indices of last non-blanks in rows 217
Indices of first blanks in rows of array X 282
Moving all blanks to end of each row 160
Moving all blanks to end of text 42
Test if X is boolean 350
Using boolean matrix Y in expanding X 469
Choosing a string according to boolean value G 493
Indexing with boolean value X (plotting a curve) 605
Zeroing all boolean values 566
Boolean vector of length Y with ones in locations X 543
Boolean vector of length Y with zeroes in locations X 539

All,	both	360
Conditional	branch depending on sign of X	119
Case structure with an encoded	branch destination	66
Conditional	branch out of program	118
Conditional	branch out of programs	89
Conditional	branch to line Y	114
X	but not Y	571
Row-	by-row formatting (width G) of X with Y decimals per row	109
Ascending	cardinal numbers (ranking, all different)	13
Descending	cardinal numbers (ranking, all different)	17
Ascending	cardinal numbers (ranking, shareable)	1
	Case structure according to key vector G	117
	Case structure with an encoded branch destination	66
	Case structure with integer switch	116
	Case structure with logical switch (preferring from start)	115
Present value of	cash flows X at interest rate Y %	77
Future value of	cash flows X at interest rate Y %	82
	Catenating G elements H after indices Y in vector X	28
	Catenating G elements H before indices Y in vector X	29
	Catenating Y elements G after every element of X	577
	Catenating Y elements G before every element of X	576
Vectors as matrices in	catenation	273
Vectors as matrices in	catenation	274
	Centering character array X with ragged edges	248
	Centering character array X with only right edge ragged	269
	Centering text line X into a field of width Y	323
	Centering text line X into a field of width Y	393
Conditional	change of elements of Y to one according to X	569
Conditional elementwise	change of sign	603
Conditional elementwise	change of sign	623
Square of elements of X without	change of sign	477
Increasing absolute value without	change of sign	475
Zeroing a vector (without	change of size)	608
	Changing an index origin dependent result to be as □IO=1	568
	Changing an index origin dependent result to be as □IO=0	625
	Changing an index origin dependent argument to act as □IO=1	626
	Changing an index origin dependent argument to act as □IO=0	628
	Changing elements in X with value Y[1] into Y[2]	380
	Changing index of an unfound element to zero	472
	Changing index of an unfound element to zero (not effective)	541
	Changing lengths of subvectors to ending indicators	277
	Changing lengths of subvectors to starting indicators	278
	Changing numeric code X into corresponding name in Y	554
	Changing starting indicators of subvectors to lengths	423
	Changing X if a new input value is given	92
Centering	character array X with only right edge ragged	269
Centering	character array X with ragged edges	248
Giving a	character default value for input	403
Rows of non-empty matrix X starting with a	character in Y	499
Dividing a 400×12	character matrix into one page	524
Sum of numbers in	character matrix X	101
Removing punctuation	characters	496
Converting	characters into numeric codes	561
Conversion of	characters to hexadecimal representation (□AV)	56
	Check for input in range 1..X	542
	Choosing a string according to boolean value G	493
	Choosing according to signum	602
Matrix for	choosing all subsets of X (truth table)	59
	Choosing sorting direction during execution	34
	Choosing Y objects out of ιX with replacement (roll)	122
	Choosing Y objects out of ιX without replacement (deal)	123
Index of first occurrence of G in X (	circularly) after Y	471
Which	class do elements of X belong to	181
Classification of elements Y into X	classes of equal size	156
Division to Y	classes with width H, minimum G	173
	Classification of elements Y into X classes of equal size	156

Representation of current time (24 hour	clock)	64
Representation of current time (12 hour	clock)	105
Rotation matrix for angle X (in radians) counter-	clockwise	132
Rounding to zero values of X	close to zero	474
Forming a transitive	closure	112
First part of numeric	code ABBB	459
Last part of numeric	code ABBB	479
Decoding numeric	code ABBCCC into a matrix	63
Changing numeric	code X into corresponding name in Y	554
Converting characters into numeric	codes	561
Sums according to	codes G	152
Annuity	coefficient: X periods at interest rate Y %	186
Pascal's triangle of order X (binomial	coefficients)	136
(Real) solution of quadratic equation with	coefficients X	363
Product of two polynomials with	coefficients X and Y	163
Value of polynomial with	coefficients Y at point X	83
Value of polynomial with	coefficients Y at point X	227
Value of Taylor series with	coefficients Y at point X	137
Value of Taylor series with	coefficients Y at point X	281
Value of polynomial with	coefficients Y at points X	69
	Cofficients of exponential (curve) fit of points (X,Y)	127
Sorting a matrix according to Y:th	column	33
First	column as a matrix	587
First	column as a vector	607
"Transpose" of matrix X with	column fields of width Y	515
Forming a two-	column matrix	589
Reshaping vector X into a one-	column matrix	534
Reshaping vector X into a one-	column matrix	594
Reshaping vector X into a two-	column matrix	591
Reshaping numeric vector X into a one-	column matrix	203
Multiplication of each	column of matrix X by vector Y	516
Adding X to each	column of Y	192
Adding X to each	column of Y	193
Alphabetizing X; equal alphabets in same	column of Y	395
Helps locating	column positions ⍳..X	46
	Column sum of a matrix	372
Removing duplicate blank	columns	220
Removing trailing blank	columns	209
Removing leading blank	columns	210
Removing blank	columns	226
Removing trailing blank	columns	272
Aligning diagonals of matrix X to	columns	399
Matrix with shape of Y and X as its	columns	517
Pairwise differences of successive	columns (inverse of +\)	383
Drop of Y first	columns from matrix X	443
Applying to	columns function F defined on rows	514
Number of	columns in array X	599
Number of	columns in matrix X	410
Number of	columns in matrix X	445
Number of	columns in matrix X	600
Average (mean value) of	columns of matrix X	326
Aligning	columns of matrix X to diagonals	398
Matrix with X	columns Y	521
Removing	columns Y from array X	396
Number of	combinations of X objects taken Y at a time	142
A way to	combine trigonometric functions (sin X cos Y)	130
Test if X and Y have elements in	common	348
Greatest	common divisor of elements of X	155
	Comparing vector Y with rows of array X	232
	Comparison of successive rows	344
	Comparison table	169
Alphabetical	comparison with alphabets G	549
Lengthening matrix X to be	compatible with Y	390
Widening matrix X to be	compatible with Y	391
	Complementary angle (arccos sin X)	131
	Compound interest for principals Y at rates G % in times X	146

(Cyclic) compression of successive blanks 494
Selection between X and Y depending on condition G 567
Index of first satisfied condition in X 559
Number of elements satisfying condition X 370
Reverse vector X on condition Y 533
Transpose matrix X on condition Y 518
Selecting elements of X satisfying condition Y 512
Selecting elements satisfying condition Y, zeroing others 622
Moving elements satisfying condition Y to the start of X 43
Conditional branch depending on sign of X 119
Conditional branch out of programs 89
Conditional branch out of program 118
Conditional branch to line Y 114
Conditional change of elements of Y to one according to X 569
Conditional drop of last element of X 448
Conditional drop of Y elements from array X 447
Conditional elementwise change of sign 603
Conditional elementwise change of sign 623
Conditional execution 96
Conditional in text 508
Transforming node matrix X into a connection matrix 157
Transforming connection matrix X into a node matrix 148
Transforming connectivity matrix X into a connectivity list 50
Transforming connectivity list X into a connectivity matrix 71
Transforming connectivity list X into a connectivity matrix 71
Transforming connectivity matrix X into a connectivity list 50
Running sum of Y consecutive elements of X 285
Consecutive integers from X to Y (arithmetic progression) 558
For-loop ending construct 113
Test if rows of X contain elements differing from Y 231
Continued fraction 619
Continuing from line Y (if X>0) or exit 120
Contour levels Y at points with altitudes X 179
Merging X, Y, Z ... under control of G (mesh) 11
Merging X and Y under control of G (mesh) 12
Merging X and Y under control of G (mesh) 16
Merging X and Y under control of G (mesh) 31
Merging X and Y under control of G (mesh) 482
Conversion from decimal to hexadecimal 49
Conversion from hexadecimal to decimal 75
Conversion of characters to hexadecimal representation (⎕AV) 56
Conversion of each row into a number (default zero) 93
Conversion of indices to indices of raveled array 81
Converting characters into numeric codes 561
Converting expression X to numeric form with default value Y 94
Test if numeric and converting into numeric form 95
Converting set of positive integers X into a mask 329
Converting zeroes to Y 574
Corner element of a (non-empty) array 615
Changing numeric code X into corresponding name in Y 554
A way to combine trigonometric functions (sin X cos Y) 130
Executing costly monadic function F on repetitive arguments 151
Index of last occurrence, counted from the rear 532
Index of first non-blank, counted from the rear 79
Rotation matrix for angle X (in radians) counter-clockwise 132
Incrementing cyclic counter X with upper limit Y 61
Vector (cross) product of vectors 528
Cumulative sum 310
Cumulative sum (+\) over subvectors of Y indicated by X 255
Restoring argument of cumulative sum (inverse of +\) 442
Playing order in a cup for X ranked players 389
Encoding current date 72
Representation of current date (American) 107
Representation of current date (ascending format) 104
Representation of current date (descending format) 65
Representation of current time (24 hour clock) 64

Representation of	current time (12 hour clock) 105
Indexing with boolean value X (plotting a	curve) 605
Predicted values of exponential (	curve) fit 124
G-degree polynomial (	curve) fit of points (X,Y) 126
Cofficients of exponential (	curve) fit of points (X,Y) 127
(	Cyclic) compression of successive blanks 494
Incrementing	cyclic counter X with upper limit Y 61
Leftmost neighboring elements	cyclically 537
Rightmost neighboring elements	cyclically 536
Entering from terminal	data exceeding input (printing) width 598
Sorting indices X according to	data Y 32
Encoding current	date 72
Representation of current	date (American) 107
Representation of current	date (ascending format) 104
Representation of current	date (descending format) 65
Separating a	date YYMMDD to YY, MM, DD 57
Number of	days in month X of year Y 74
Separating a date YYMMDD to YY, MM,	DD 57
Choosing Y objects out of ιX without replacement (	deal) 123
Conversion from hexadecimal to	decimal 75
Binary format of	decimal number X 45
Conversion from	decimal to hexadecimal 49
Rounding to X	decimals 460
Arithmetic precision of the system (in	decimals) 450
Number of	decimals (up to Y) of elements of X 149
Formatting X with H	decimals in fields of width G 110
Number of	decimals of elements of X 87
Row-by-row formatting (width G) of X with Y	decimals per row 109
	Decoding numeric code ABBCCC into a matrix 63
	Decommenting a matrix representation of a function (⎕CR) 258
Fifo stack Y	decremented with X units 286
Giving a numeric	default value for input 97
Giving a character	default value for input 403
Limiting indices and giving a	default value G 458
Converting expression X to numeric form with	default value Y 94
Conversion of each row into a number (	default zero) 93
Applying to columns function F	defined on rows 514
	Definite integral of F(X) in range Y with G steps ('X'∊F) 91
G-	degree polynomial (curve) fit of points (X,Y) 126
Transformation from radians to	degrees 133
Transformation from	degrees to radians 134
Kronecker	delta of X and Y (element of identity matrix) 575
Changing an index origin	dependent argument to act as ⎕IO=1 626
Changing an index origin	dependent argument to act as ⎕IO=0 628
Changing an index origin	dependent result to be as ⎕IO=1 568
Changing an index origin	dependent result to be as ⎕IO=0 625
Selection between X and Y	depending on condition G 567
Selection	depending on index origin 604
Selection	depending on index origin 609
Conditional branch	depending on sign of X 119
Zero or space	depending on the type of X (fill element) 409
Zeroing elements of Y	depending on their values 545
Number of digit positions in scalar X (	depends on ⎕PP) 111
	Depth of parentheses 295
	Derivate of polynomial X 430
	Descending cardinal numbers (ranking, all different) 17
Representation of current date (	descending format) 65
Sorting X into	descending order 44
Test if X is in	descending order along direction Y 270
Case structure with an encoded branch	destination 66
Evaluating a two-row	determinant 313
Evaluating a two-row	determinant 314
Sum over elements of X	determined by elements of Y 150
Standard	deviation of X 319
	Diagonal elements of matrix X 526
	Diagonal matrix with elements of X 429

Main diagonal of array X 525
Aligning columns of matrix X to diagonals 398
Aligning diagonals of matrix X to columns 399
Elements of X not in Y (difference of sets) 498
Which elements of X are not in set Y (difference of sets) 552
Pairwise differences of successive columns (inverse of +\) 383
Differences of successive elements of X along direction Y 436
Pairwise differences of successive elements of vector X 420
Ascending cardinal numbers (ranking, all different) 13
Descending cardinal numbers (ranking, all different) 17
Index of first differing element in vectors X and Y 551
Test if rows of X contain elements differing from Y 231
Test if an element differs from the next one 431
Test if an element differs from the previous one 432
Number of digit positions in integers in X 455
Number of digit positions in scalar X (depends on ⎕PP) 111
Number of digitpositions in integers in X 452
Number of digits in positive integers in X 456
Digits of X separately 53
Adding a new dimension after dimension G Y-fold 10
Adding a new dimension after dimension G Y-fold 10
Replicating a dimension of rank three array X Y-fold 200
Reduction in dimension Y, rank unchanged 502
Direct matrix product 187
Choosing sorting direction during execution 34
Test if X is in ascending order along direction Y 268
Test if X is in descending order along direction Y 270
Average (mean value) of elements of X along direction Y 364
Differences of successive elements of X along direction Y 436
Variance (dispersion) of X 320
Extending a distance table to next leg 224
Histogram (distribution barchart) down the page) 144
Poisson distribution of states X with average number Y 140
Distribution of X into intervals between Y 145
Binomial distribution of X trials with probability Y 138
Dividing a 400×12 character matrix into one page 524
Divisibility table 164
Elements of X divisible by Y 470
Force 0÷0 into DOMAIN ERROR in division 620
Avoiding division by zero error (gets value zero) 572
Division to Y classes with width H, minimum G 173
Greatest common divisor of elements of X 155
Divisors of X 468
Force 0÷0 into DOMAIN ERROR in division 620
Scalar (dot) product of vectors 243
Doubling quotes (for execution) 25
Grading down subvectors of Y indicated by X 15
Barchart of integer values (down the page) 166
Histogram (distribution barchart) down the page) 144
Formatting a numerical vector to run down the page 103
Conditional drop of last element of X 448
Conditional drop of Y elements from array X 447
Drop of Y first columns from matrix X 443
Drop of Y first rows from matrix X 388
Drop of Y first rows from matrix X 444
Removing duplicate blank columns 220
Removing duplicate blank rows 218
Removing duplicate elements (nub) 154
Removing duplicate elements (nub) 501
Removing duplicate rows 70
Removing duplicate rows 206
Removing duplicate rows 208
Removing duplicate rows from ordered matrix X 327
Removing duplicate Y's from vector X 426
Removing duplicates from an ordered vector 422
Removing duplicates from positive integers 505

	Duplicating element of X belonging to Y,1↑X until next found
	Duplicating vector X Y times 610
Choosing sorting direction	during execution 34
Multiplication of	each column of matrix X by vector Y 516
Adding X to	each column of Y 192
Adding X to	each column of Y 193
Inserting Y after	each element of X 581
Test if	each element of X occurs only once 353
Vector to expand a new element after	each one in X 507
Test if X and Y are permutations of	each other 4
Moving all blanks to end of	each row 160
Conversion of	each row into a number (default zero) 93
Multiplication of	each row of X by vector Y 611
Adding X to	each row of Y 189
Adding X to	each row of Y 190
Transferring text X to right	edge of field of width Y 412
Centering character array X with only right	edge ragged 269
Centering character array X with ragged	edges 248
	Editing X with Y ∇-wise 263
Changing index of an unfound element to zero (not	effective) 541
Insert vector for X[i] elements before i:th	element 279
Zero or space depending on the type of X (fill	element) 409
Vector to expand a new	element after each one in X 507
Test if an	element differs from the next one 431
Test if an	element differs from the previous one 432
Index of first differing	element in vectors X and Y 551
Corner	element of a (non-empty) array 615
Kronecker delta of X and Y (	element of identity matrix) 575
Removing every second	element of X 469
Inserting Y after each	element of X 581
Removing every Y:th	element of X 466
Taking every Y:th	element of X 467
Index of (first) minimum	element of X 22
Index of (first) minimum	element of X 336
Index of (first) maximum	element of X 23
Index of (first) maximum	element of X 330
Conditional drop of last	element of X 448
Adding scalar Y to last	element of X 406
Catenating Y elements G after every	element of X 577
Catenating Y elements G before every	element of X 576
Locating field number Y starting with first	element of X 283
Duplicating	element of X belonging to Y,1↑X until next found 297
Test if each	element of X occurs only once 353
Filling X with last	element of X to length Y 377
Replacing last	element of X with Y 433
Replacing first	element of X with Y 434
Index of last	element of Y in X 530
Changing index of an unfound	element to zero 472
Changing index of an unfound	element to zero (not effective) 541
Reshaping one-	element vector X into a scalar 371
Indexing scattered	elements 80
Rightmost neighboring	elements 384
Leftmost neighboring	elements 385
Lengths of subvectors with equal	elements 413
Rotate first	elements (1⊖) of subvectors of Y indicated by X 21
Number of	elements (also of a scalar) 366
Number of	elements (also of a scalar) 596
Removing duplicate	elements (nub) 154
Removing duplicate	elements (nub) 501
Insert vector for X[i]	elements before i:th element 279
Rightmost neighboring	elements cyclically 536
Leftmost neighboring	elements cyclically 537
Test if rows of X contain	elements differing from Y 231
Selecting specific	elements from a 'large' outer product 194
Conditional drop of Y	elements from array X 447
Take of at most X	elements from Y 394

Catenating Y elements G after every element of X 577
Catenating Y elements G before every element of X 576
Catenating G elements H after indices Y in vector X 28
Catenating G elements H before indices Y in vector X 29
Test if X and Y have elements in common 348
Changing elements in X with value Y[1] into Y[2] 380
All pairs of elements of ⍳X and ⍳Y 58
'X⍳Y' by rows for elements of matrices 153
Diagonal elements of matrix X 526
Maxima of elements of subsets of X specified by Y 213
Minima (⌊/) of elements of subvectors of Y indicated by X 6
Maxima (⌈/) of elements of subvectors of Y indicated by X 14
Pairwise ratios of successive elements of vector X 419
Pairwise differences of successive elements of vector X 420
Test if all elements of vector X are equal 159
Test if all elements of vector X are equal 214
Test if all elements of vector X are equal 215
Test if all elements of vector X are equal 311
Test if all elements of vector X are equal 315
Test if all elements of vector X are equal 316
Test if all elements of vector X are equal 317
Test if all elements of vector X are equal 324
Test if all elements of vector X are equal 340
Test if all elements of vector X are equal 341
Test if all elements of vector X are equal 343
Test if all elements of vector X are equal 347
Test if all elements of vector X are equal 349
Test if all elements of vector X are equal 352
Test if all elements of vector X are equal 354
Test if all elements of vector X are equal 358
Size of range of elements of X 312
Minimum of elements of X 339
Maximum of elements of X 335
Sum of elements of X 374
Greatest common divisor of elements of X 155
Number of decimals of elements of X 87
Product of elements of X 368
Diagonal matrix with elements of X 429
Running sum of Y consecutive elements of X 285
Number of decimals (up to Y) of elements of X 149
Numeric headers (elements of X) for rows of table Y 102
Differences of successive elements of X along direction Y 436
Average (mean value) of elements of X along direction Y 364
Which elements of X are not in set Y (difference of sets) 552
Which class do elements of X belong to 181
Sum over elements of X determined by elements of Y 150
Elements of X divisible by Y 470
Replacing elements of X in set Y by blanks / zeroes 480
Sum elements of X marked by succeeding identicals in Y 284
Replacing elements of X not fulfilling Y by blanks / zeroes 483
Replacing elements of X not in set Y by blanks / zeroes 481
Elements of X not in Y (difference of sets) 498
Selecting elements of X satisfying condition Y 512
Square of elements of X without change of sign 477
Index of first occurrence of elements of Y 338
Sum over elements of X determined by elements of Y 150
Zeroing elements of Y depending on their values 545
Indices of all occurrences of elements of Y in X 495
Replacing elements of Y satisfying X with G 504
Conditional change of elements of Y to one according to X 569
Number of elements satisfying condition X 370
Selecting elements satisfying condition Y, zeroing others 622
Moving elements satisfying condition Y to the start of X 43
Removing elements Y from beginning and end of vector X 416
Removing elements Y from beginning of vector X 435
Removing elements Y from end of vector X 427

Retrieving scattered elements Y from matrix X 522
Removing elements Y from vector X 509
Indices of elements Y in array X 55
Classification of elements Y into X classes of equal size 156
Conditional elementwise change of sign 603
Conditional elementwise change of sign 623
Corner element of a (non-empty) array 615
Test if X is an empty array 548
Arithmetic average (mean value), also for an empty array 332
Forming an initially empty array to be expanded 618
Output of an empty line 617
Reshaping non-empty lower-rank array X into a matrix 392
Reshaping non-empty lower-rank array X into a matrix 402
Empty matrix 513
Rows of non-empty matrix X starting with a character in Y 499
Empty numeric vector 563
Adding an empty row into X after row Y 487
Adding an empty row into X after rows Y 485
Positive maximum, at least zero (also for empty X) 334
Case structure with an encoded branch destination 66
Encoding current date 72
Moving all blanks to end of each row 160
Inserting vector Y to the end of matrix X 376
Moving all blanks to end of text 42
Removing elements Y from end of vector X 427
Removing elements Y from beginning and end of vector X 416
Scan from end with α 484
For-loop ending construct 113
Changing lengths of subvectors to ending indicators 277
Field lengths of vector X; G ↔ ending indices 414
Ending points for X fields of width Y 404
Ending points for X in indices pointed by Y 529
Ending points of equal groups 417
Entering from terminal data exceeding input (printing) width 598
Test if all elements of vector X are equal 159
Test if all elements of vector X are equal 214
Test if all elements of vector X are equal 215
Test if all elements of vector X are equal 311
Test if all elements of vector X are equal 315
Test if all elements of vector X are equal 316
Test if all elements of vector X are equal 317
Test if all elements of vector X are equal 324
Test if all elements of vector X are equal 340
Test if all elements of vector X are equal 341
Test if all elements of vector X are equal 343
Test if all elements of vector X are equal 347
Test if all elements of vector X are equal 349
Test if all elements of vector X are equal 352
Test if all elements of vector X are equal 354
Test if all elements of vector X are equal 358
Alphabetizing X; equal alphabets in same column of Y 395
Lengths of subvectors with equal elements 413
Ending points of equal groups 417
Starting points of equal groups 418
Test if arrays of equal shape are identical 357
Classification of elements Y into X classes of equal size 156
(Real) solution of quadratic equation with coefficients X 363
Avoiding division by zero error (gets value zero) 572
Force 0+0 into DOMAIN ERROR in division 620
Syntax error to stop execution 629
Evaluating a two-row determinant 313
Evaluating a two-row determinant 314
Evaluation of several expressions; results form a vector 99
Test if even 473
Rounding to nearest even integer 453
Rounding, to nearest even integer for .5 = 1||X 454

Catenating Y elements G after	*every element of X*	577
Catenating Y elements G before	*every element of X*	576
Removing	*every second element of X*	469
Taking	*every Y:th element of X*	467
Removing	*every Y:th element of X*	466
Entering from terminal data	*exceeding input (printing) width*	598
	Exclusive or	573
	Executing costly monadic function F on repetitive arguments	151
Conditional	*execution*	96
Syntax error to stop	*execution*	629
Doubling quotes (for	*execution)*	25
Choosing sorting direction during	*execution*	34
Continuing from line Y (if X>0) or	*exit*	120
Vector to	*expand a new element after each one in X*	507
Forming first row of a matrix to be	*expanded*	408
Forming an initially empty array to be	*expanded*	618
Using boolean matrix Y in	*expanding X*	489
	Expansion vector with zero after indices Y	27
	Expansion vector with zero after indices Y	538
Predicted values of	*exponential (curve) fit*	124
Cofficients of	*exponential (curve) fit of points (X,Y)*	127
Setting latent	*expression*	631
Converting	*expression X to numeric form with default value Y*	94
Evaluation of several	*expressions; results form a vector*	99
Assign values of	*expressions in X to variables named in Y*	98
	Extending a distance table to next leg	224
	Extending a transitive binary relation	230
	Field lengths of vector X1 G ↔ ending indices	414
Locating	*field number Y starting with first element of X*	283
Centering text line X into a	*field of width Y*	323
Centering text line X into a	*field of width Y*	393
Transferring text X to right edge of	*field of width Y*	412
Limits X to fit in ⍕	*field Y[1 2]*	212
Formatting X with H decimals in	*fields of width G*	110
Ending points for X	*fields of width Y*	404
Starting points for X	*fields of width Y*	405
"Transpose" of matrix X with column	*fields of width Y*	515
Leading zeroes for positive numbers X in	*fields of width Y*	106
Justifying left	*fields of X (lengths Y) to length G*	185
Justifying right	*fields of X (lengths Y) to length G*	184
	Fifo stock Y decremented with X units	286
X first	*figurate numbers*	260
Zero or space depending on the type of X (	*fill element)*	409
	Filling X with last element of X to length Y	377
Index of (	*first) maximum element of X*	23
Index of (	*first) maximum element of X*	330
Index of (	*first) minimum element of X*	22
Index of (	*first) minimum element of X*	336
Indices of	*first blanks in rows of array X*	282
	First column as a matrix	567
	First column as a vector	607
Drop of Y	*first columns from matrix X*	443
Index of	*first differing element in vectors X and Y*	551
Locating field number Y starting with	*first element of X*	283
Replacing	*first element of X with Y*	434
Rotate	*first elements (1⊖) of subvectors of Y indicated by X*	21
X	*first figurate numbers*	260
	First group of ones	261
Index of	*first non-blank, counted from the rear*	79
Index of	*first occurrence of elements of Y*	338
Index of	*first occurrence of G in X (circularly) after Y*	471
	First occurrence of string X in string Y	178
	First occurrence of string X in matrix Y	229
Turn on all zeroes after	*first one*	304
Turn off all ones after	*first one*	307
Index of	*first one after index Y in X*	438

	First ones in groups of ones 381
	First ones in groups of ones 440
	First ones of subvectors of Y indicated by X (<\) 421
	First part of numeric code ABBB 459
Move set of points X into	*first quadrant 174*
Forming	*first row of a matrix to be expanded 408*
Drop of Y	*first rows from matrix X 388*
Drop of Y	*first rows from matrix X 444*
Index of	*first satisfied condition in X 559*
X	*first triangular numbers 302*
	First word in X 401
Turn off all ones after	*first zero 306*
Turn on all zeroes after	*first zero 308*
Predicted values of exponential (curve)	*fit 124*
Predicted values of best linear	*fit (least squares) 125*
Limits X to	*fit in ⍕ field Y[1 2] 212*
G-degree polynomial (curve)	*fit of points (X,Y) 126*
Cofficients of exponential (curve)	*fit of points (X,Y) 127*
Best linear	*fit of points (X,Y) (least squares) 128*
Present value of cash	*flows X at interest rate Y % 77*
Future value of cash	*flows X at interest rate Y % 82*
Adding a new dimension after dimension G Y-	*fold 10*
Replicating a dimension of rank three array X Y-	*fold 200*
	For-loop ending construct 113
	Force 0÷0 into DOMAIN ERROR in division 520
Test if numeric and converting into numeric	*form 95*
Evaluation of several expressions' results	*form a vector 99*
Converting expression X to numeric	*form with default value Y 94*
Representation of current date (ascending	*format) 104*
Representation of current date (descending	*format) 65*
Binary	*format of decimal number X 45*
Row-by-row	*formatting (width G) of X with Y decimals per row 109*
	Formatting a numerical vector to run down the page 103
	Formatting with zero values replaced by blanks 108
	Formatting X with H decimals in fields of width G 110
	Forming a transitive closure 112
	Forming a two-column matrix 589
	Forming a two-row matrix 588
	Forming a Y-row matrix with all rows alike (X) 593
	Forming an initially empty array to be expanded 618
	Forming first row of a matrix to be expanded 408
Area of triangle with side lengths in X (Heron's	*formula) 318*
Duplicating element of X belonging to Y,1↑X until next	*found 297*
Continued	*fraction 619*
	Fractional part 478
Magnitude of	*fractional part 465*
	Fractional part with sign 476
Integer and	*fractional parts of positive numbers 62*
	Framing 464
Replacing elements of X not	*fulfilling Y by blanks / zeroes 483*
Gamma	*function 141*
Beta	*function 139*
Decommenting a matrix representation of a	*function (⎕CR) 258*
Applying to columns	*function F defined on rows 514*
Executing costly monadic	*function F on repetitive arguments 151*
A way to combine trigonometric	*functions (sin X cos Y) 130*
	Future value of cash flows X at interest rate Y % 82
	Gamma function 141
Open a	*gap of X[i] after Y[G[i]] (for all i) 264*
Open a	*gap of X[i] before Y[G[i]] (for all i) 271*
Avoiding division by zero error (	*gets value zero) 572*
Changing X if a new input value is	*given 92*
	Giving a character default value for input 403
Limiting indices and	*giving a default value G 458*
	Giving a numeric default value for input 97
	Grade up according to key Y 30

Grading down subvectors of Y indicated by X 15
Grading up subvectors of Y indicated by X 7
Graph of F(X) at points X ('X'<F) 90
Greatest common divisor of elements of X 155
First group of ones 261
Ending points of equal groups 417
Starting points of equal groups 418
First ones in groups of ones 381
Last ones in groups of ones 439
First ones in groups of ones 440
Lengths of groups of ones in X 415
Groups of ones in Y pointed by X (or trailing parts) 256
Groups of ones in Y pointed by X 292
Handling array X temporarily as a vector 592
Test if X and Y have elements in common 348
Reversal (⌽) of subvectors having lengths Y 39
Numeric headers (elements of X) for rows of table Y 102
Barchart of X with height Y (across the page) 170
Avoiding parentheses with help of reversal 535
Helps locating column positions 1..X 46
Area of triangle with side lengths in X (Heron's formula) 318
Conversion from decimal to hexadecimal 49
Conversion of characters to hexadecimal representation (⎕AV) 56
Conversion from hexadecimal to decimal 75
Hilbert matrix of order X 198
Histogram (distribution barchart) down the page) 144
Representation of current time (24 hour clock) 64
Representation of current time (12 hour clock) 105
Rounding to nearest hundredth 461
Test if arrays are identical 544
Test if arrays of equal shape are identical 357
Sum elements of X marked by succeeding identicals in Y 284
X=X identity matrix 197
X=X identity matrix 582
Kronecker delta of X and Y (element of identity matrix) 575
Identity of two sets 345
X implies Y 570
Locations of texts between and including quotes 287
Locations of texts between and including quotes 293
Limiting X between Y[1] and Y[2], inclusive 449
Increasing absolute value without change of sign 475
Increasing rank of Y to rank of X 590
Incrementing cyclic counter X with upper limit Y 61
Indexing independent of index origin 606
Inserting X into Y after index G 382
Index of (first) maximum element of X 23
Index of (first) maximum element of X 330
Index of (first) minimum element of X 22
Index of (first) minimum element of X 336
Changing index of an unfound element to zero 472
Changing index of an unfound element to zero (not effective) 541
Index of first differing element in vectors X and Y 551
Index of first non-blank, counted from the rear 79
Index of first occurrence of elements of Y 338
Index of first occurrence of G in X (circularly) after Y 471
Index of first one after index Y in X 438
Index of first satisfied condition in X 559
Index of key Y in key vector X 562
Index of last element of Y in X 530
Index of last occurrence 531
Index of last occurrence, counted from the rear 532
Moving index of width Y for vector X 201
Indexing Independent of index origin 606
Selection depending on index origin 604
Selection depending on index origin 609
Index origin (⎕IO) as a vector 560

Changing an index origin dependent argument to act as ⎕IO=1 626
Changing an index origin dependent argument to act as ⎕IO=0 628
Changing an index origin dependent result to be as ⎕IO=1 568
Changing an index origin dependent result to be as ⎕IO=0 625
Index pairs of saddle points 48
Index of first one after index Y in X 438
Indexing independent of index origin 606
Indexing scattered elements 80
Indexing when rank is not known beforehand 100
Indexing with boolean value X (plotting a curve) 605
Temporary ravel of X for indexing with G 584
Temporary ravel of X for indexing with G 585
Grading up subvectors of Y indicated by X 7
Grading down subvectors of Y indicated by X 15
First ones of subvectors of Y indicated by X (<\) 421
Sums over (+/) subvectors of Y indicated by X 291
G:th subvector of Y (subvectors indicated by X) 300
Or-scan (∨\) over subvectors of Y indicated by X 289
And-scan (∧\) over subvectors of Y indicated by X 290
Or-reduction (∨/) of subvectors of Y indicated by X 491
And-reduction (∧/) of subvectors of Y indicated by X 492
Maximum scan (⌈\) over subvectors of Y indicated by X 2
Minimum scan (⌊\) over subvectors of Y indicated by X 3
Cumulative sum (+\) over subvectors of Y indicated by X 255
Running parity (≠\) over subvectors of Y indicated by X 254
Maxima (⌈/) of elements of subvectors of Y indicated by X 14
Minima (⌊/) of elements of subvectors of Y indicated by X 6
Rotate first elements (1⊖) of subvectors of Y indicated by X 21
Changing lengths of subvectors to ending indicators 277
Changing lengths of subvectors to starting indicators 278
Changing starting indicators of subvectors to lengths 423
Field lengths of vector X; G ↔ ending indices 414
Limiting indices and giving a default value G 458
Inserting Y *'s into vector X after indices G 26
Indices of all occurrences of elements of Y in X 495
Indices of all occurrences of Y in X 503
Indices of elements Y in array X 55
Indices of first blanks in rows of array X 282
Indices of last non-blanks in rows 217
Indices of ones in logical vector X 41
Indices of ones in logical vector X 280
Indices of ones in logical vector X 506
Conversion of indices to indices of raveled array 81
Indices of subvectors of length Y starting at X+1 202
All indices of vector X 556
Matrix of all indices of X 51
Ending points for X in indices pointed by Y 529
Starting points for X in indices pointed by Y 540
Conversion of indices to indices of raveled array 81
Reversal (⊖) of subvectors starting at indices X 40
Sorting indices X according to data Y 32
Expansion vector with zero after indices Y 27
Expansion vector with zero after indices Y 538
Catenating G elements H after indices Y in vector X 28
Catenating G elements H before indices Y in vector X 29
Ordinal numbers of words in X that indices Y point to 177
Positive infinity; the largest representable value 337
Negative infinity; the smallest representable value 331
Forming an initially empty array to be expanded 618
Giving a numeric default value for input 97
Giving a character default value for input 403
Entering from terminal data exceeding input (printing) width 598
Check for input in range 1..X 542
Input of row Y of text matrix X 378
Changing X if a new input value is given 92
Insert vector for X[i] elements before i:th element 279

Insert vector for X[i] zeroes after i:th subvector 265
Inserting vector X into matrix Y after row G 375
Inserting vector Y to the end of matrix X 376
Inserting X into Y after index G 382
Inserting Y *'s into vector X after indices G 26
Inserting Y after each element of X 561
Test if integer 457
Rounding to nearest even integer 453
Rounding to nearest integer 462
Integer and fractional parts of positive numbers 62
Rounding, to nearest even integer for .5 = 1[|X 454
Integer representation of logical vectors 84
Barchart of two integer series (across the page) 68
Case structure with integer switch 116
Barchart of integer values (across the page) 171
Barchart of integer values (down the page) 166
Test if X is an integer within range [G,H) 221
Quick membership (∊) for positive integers 333
Removing duplicates from positive integers 505
Consecutive integers from X to Y (arithmetic progression) 558
Number of digitpositions in integers in X 452
Number of digits in positive integers in X 456
Number of digit positions in integers in X 455
Converting set of positive integers X into a mask 329
Definite integral of F(X) in range Y with G steps ('X'∊F) 91
Compound interest for principals Y at rates G % in times X 146
Future value of cash flows X at interest rate Y % 82
Present value of cash flows X at interest rate Y % 77
Annuity coefficient: X periods at interest rate Y % 186
Interpolated value of series (X,Y) at G 67
Interpretation of roman numbers 342
Intersection of sets, ∩ 500
Distribution of X into intervals between Y 145
Restoring argument of cumulative sum (inverse of +\) 442
Pairwise differences of successive columns (inverse of +\) 383
Inverting a permutation 37
Inverting a permutation 550
Inverting a permutation 553
Joining pairs of ones 303
Joining sentences 597
Justifying left 276
Justifying left fields of X (lengths Y) to length G 185
Justifying right 76
Justifying right 275
Justifying right fields of X (lengths Y) to length G 184
Case structure according to key vector G 117
Index of key Y in key vector X 562
Grade up according to key Y 30
Sorting rows of X according to key Y (alphabetizing) 18
Index of key Y in key vector X 562
Indexing when rank is not known beforehand 100
Kronecker delta of X and Y (element of identity matrix) 575
Selecting specific elements from a 'large' outer product 194
Positive infinity; the largest representable value 337
Conditional drop of last element of X 448
Adding scalar Y to last element of X 406
Filling X with last element of X to length Y 377
Replacing last element of X with Y 433
Index of last element of Y in X 530
Indices of last non-blanks in rows 217
Index of last occurrence 531
Index of last occurrence, counted from the rear 532
Last ones in groups of ones 439
Last part of numeric code ABBB 479
Setting latent expression 631
Removing leading, multiple and trailing Y's 379

Removing	*leading and trailing blanks*	259
Removing	*leading blank columns*	210
Removing	*leading blank rows*	211
Removing	*leading blanks*	267
Removing	*leading blanks*	299
Matrix with X[ι]	*leading ones on row i*	172
Removing	*leading zeroes*	437
	Leading zeroes for positive numbers X in fields of width Y	106
Matrix with X[i]	*leading zeroes on row i*	168
Test if X is a	*leap year*	463
Best linear fit of points (X,Y) (	*least squares)*	128
Predicted values of best linear fit (	*least squares)*	125
Positive maximum, at	*least zero (also for empty X)*	334
Justifying	*left*	276
Vector of length Y with X ones on the	*left, the rest zeroes*	407
Justifying	*left fields of X (lengths Y) to length G*	185
Shifting vector X	*left with Y without rotate*	387
	Leftmost neighboring elements	385
	Leftmost neighboring elements cyclically	537
Extending a distance table to next	*leg*	224
Justifying left fields of X (lengths Y) to	*length G*	185
Justifying right fields of X (lengths Y) to	*length G*	184
Separating variable	*length lines*	579
Filling X with last element of X to	*length Y*	377
Indices of subvectors of	*length Y starting at X+1*	202
Boolean vector of	*length Y with ones in locations X*	543
Vector of	*length Y with X ones on the left, the rest zeroes*	407
Boolean vector of	*length Y with zeroes in locations X*	539
	Lengthening matrix X to be compatible with Y	390
Changing starting indicators of subvectors to	*lengths*	423
Starting positions of subvectors (	*lengths in X)*	296
Area of triangle with side	*lengths in X (Heron's formula)*	318
Sums over (+/) subvectors of X,	*lengths in Y*	257
	Lengths of groups of ones in X	415
Changing	*lengths of subvectors to ending indicators*	277
Changing	*lengths of subvectors to starting indicators*	275
	Lengths of subvectors with equal elements	413
Field	*lengths of vector X; G ↔ ending indices*	414
Sorting subvectors of	*lengths Y*	5
Reversal (⌽) of subvectors having	*lengths Y*	39
Justifying left fields of X (	*lengths Y) to length G*	185
Justifying right fields of X (	*lengths Y) to length G*	184
Contour	*levels Y at points with altitudes X*	179
Sorting a matrix into	*lexicographic order*	19
Sorting a matrix into reverse	*lexicographic order*	38
Incrementing cyclic counter X with upper	*limit Y*	61
	Limiting indices and giving a default value G	458
	Limiting X between Y[1] and Y[2], inclusive	449
	Limits X to fit in ⍕ field Y[1 2]	212
Output of an empty	*line*	617
Centering text	*line X into a field of width Y*	323
Centering text	*line X into a field of width Y*	393
Conditional branch to	*line Y*	114
Continuing from	*line Y (if X>0) or exit*	120
Predicted values of best	*linear fit (least squares)*	125
Best	*linear fit of points (X,Y) (least squares)*	128
Separating variable length	*lines*	579
	List of names in X (one per row)	441
Transforming connectivity matrix X into a connectivity	*list*	50
Transforming connectivity	*list X into a connectivity matrix*	71
Helps	*locating column positions 1..X*	46
	Locating field number Y starting with first element of X	283
	Locations of texts between and including quotes	287
	Locations of texts between and including quotes	293
	Locations of texts between quotes	288
	Locations of texts between quotes	294

Boolean vector of length Y with ones in locations X 543
Boolean vector of length Y with zeroes in locations X 539
Case structure with logical switch (preferring from start) 115
Indices of ones in logical vector X 41
Indices of ones in logical vector X 280
Indices of ones in logical vector X 506
Integer representation of logical vectors 84
For-loop ending construct 113
Reshaping non-empty lower-rank array X into a matrix 392
Reshaping non-empty lower-rank array X into a matrix 402
Test if X is a lower triangular matrix 162
X×X lower triangular matrix 196
Magnitude of fractional part 465
Main diagonal of array X 525
Sum elements of X marked by succeeding identicals in Y 284
Converting set of positive integers X into a mask 329
'X⍳Y' for rows of matrices 207
'X⍳Y' by rows for elements of matrices 153
Vectors as matrices in catenation 273
Vectors as matrices in catenation 274
Column sum of a matrix 372
Row sum of a matrix 373
First column as a matrix 587
Forming a two-column matrix 589
X×X upper triangular matrix 195
X×X lower triangular matrix 196
Repeat matrix 182
Forming a two-row matrix 588
X×X identity matrix 197
X×X identity matrix 582
Empty matrix 513
Test if X is a lower triangular matrix 162
Test if X is an upper triangular matrix 161
Reshaping vector X into a one-row matrix 595
Decoding numeric code ABBCCC into a matrix 63
Reshaping vector X into a one-column matrix 534
Reshaping vector X into a one-column matrix 594
Reshaping vector X into a two-column matrix 591
Reshaping numeric vector X into a one-column matrix 203
Transforming connection matrix X into a node matrix 148
Transforming node matrix X into a connection matrix 157
Reshaping non-empty lower-rank array X into a matrix 392
Reshaping non-empty lower-rank array X into a matrix 402
Kronecker delta of X and Y (element of identity matrix) 575
Sorting a matrix according to Y:th column 33
Rotation matrix for angle X (in radians) counter-clockwise 132
Matrix for choosing all subsets of X (truth table) 59
Sorting a matrix into lexicographic order 19
Dividing a 400×12 character matrix into one page 524
Sorting a matrix into reverse lexicographic order 38
'Position' matrix of a vector 143
Matrix of all indices of X 51
Hilbert matrix of order X 198
Matrix product 240
Direct matrix product 187
Decommenting a matrix representation of a function (⎕CR) 258
Matrix sum 191
Forming first row of a matrix to be expanded 408
Transforming connectivity list X into a connectivity matrix 71
Forming a Y-row matrix with all rows alike (X) 593
Diagonal matrix with elements of X 429
Matrix with shape of Y and X as its columns 517
Matrix with X columns Y 521
Matrix with X[i] leading ones on row i 172
Matrix with X[i] leading zeroes on row i 168
Matrix with X[i] trailing ones on row i 167

	Matrix with X[i] trailing zeroes on row i	165
Diagonal elements of	matrix X	526
Test for symmetricity of	matrix X	86
Drop of Y first rows from	matrix X	388
Drop of Y first rows from	matrix X	444
Number of columns in	matrix X	410
Number of columns in	matrix X	445
Number of columns in	matrix X	600
Number of rows in	matrix X	411
Number of rows in	matrix X	446
Number of rows in	matrix X	601
Number of rows in	matrix X	613
Sum of numbers in character	matrix X	101
Input of row Y of text	matrix X	378
Drop of Y first columns from	matrix X	443
Test for antisymmetricity of	matrix X	85
Average (mean value) of rows of	matrix X	325
Test for symmetricity of square	matrix X	520
Inserting vector Y to the end of	matrix X	376
Average (mean value) of columns of	matrix X	326
Test for antisymmetricity of square	matrix X	519
Removing duplicate rows from ordered	matrix X	327
Retrieving scattered elements Y from	matrix X	522
Multiplication of each column of	matrix X by vector Y	516
Transforming node	matrix X into a connection matrix	157
Transforming connectivity	matrix X into a connectivity list	50
Transforming connection	matrix X into a node matrix	148
Sorting rows of	matrix X into ascending order	8
Sorting rows of	matrix X into ascending order	9
Transpose	matrix X on condition Y	518
Rows of non-empty	matrix X starting with a character in Y	499
Rows of	matrix X starting with string Y	216
Widening	matrix X to be compatible with Y	391
Lengthening	matrix X to be compatible with Y	390
Aligning diagonals of	matrix X to columns	399
Aligning columns of	matrix X to diagonals	398
"Transpose" of	matrix X with column fields of width Y	515
First occurrence of string X in	matrix Y	229
Inserting vector X into	matrix Y after row G	375
Using boolean	matrix Y in expanding X	489
	Maxima (⌈/) of elements of subvectors of Y indicated by X	14
	Maxima of elements of subsets of X specified by Y	213
Positive	maximum, at least zero (also for empty X)	334
Index of (first)	maximum element of X	23
Index of (first)	maximum element of X	330
	Maximum of elements of X	335
	Maximum of X with weights Y	222
	Maximum scan (⌈\) over subvectors of Y indicated by X	2
	Maximum table	183
Arithmetic average (	mean value)	365
Arithmetic average (	mean value), also for an empty array	332
Average (	mean value) of columns of matrix X	326
Average (	mean value) of elements of X along direction Y	364
Average (	mean value) of rows of matrix X	325
Arithmetic average (	mean value) of X weighted by Y	237
	Median	24
Quick	membership (∊) for positive integers	333
	Merging vectors X and Y alternately	578
	Merging X, Y, Z ... under control of G (mesh)	11
	Merging X and Y under control of G (mesh)	12
	Merging X and Y under control of G (mesh)	16
	Merging X and Y under control of G (mesh)	31
	Merging X and Y under control of G (mesh)	482
Merging X and Y under control of G (	mesh)	12
Merging X and Y under control of G (	mesh)	16
Merging X and Y under control of G (	mesh)	31

Merging X and Y under control of G (	mesh)	482
Merging X, Y, Z ... under control of G (	mesh)	11
	Minima (⌊/) of elements of subvectors of Y indicated by X	6
Index of (first)	minimum element of X	22
Index of (first)	minimum element of X	336
Division to Y classes with width H,	minimum G	173
	Minimum of elements of X	339
	Minimum of X with weights Y	223
	Minimum scan (⌊\) over subvectors of Y indicated by X	3
Array and its negative ('plus	minus')	204
Array and its negative ('plus	minus')	583
Number and its negative ('plus	minus')	621
Separating a date YYMMDD to YY,	MM, DD	57
Y:th	moment of X	321
Executing costly	monadic function F on repetitive arguments	151
Number of days in	month X of year Y	74
Take of at	most X elements from Y	394
	Move set of points X into first quadrant	174
	Moving all blanks to end of each row	160
	Moving all blanks to end of text	42
	Moving elements satisfying condition Y to the start of X	43
	Moving index of width Y for vector X	201
Removing leading,	multiple and trailing Y's	379
Removing	multiple blanks	424
	Multiplication of each column of matrix X by vector Y	516
	Multiplication of each row of X by vector Y	611
	Multiplication table	199
Test if X is a valid APL	name	322
Changing numeric code X into corresponding	name in Y	554
Using a variable	named according to X	88
Assign values of expressions in X to variables	named in Y	98
List of	names in X (one per row)	441
Rounding to	nearest even integer	453
Rounding, to	nearest even integer for .5 = 1||X	454
Rounding to	nearest hundredth	461
Rounding to	nearest integer	462
Array and its	negative ('plus minus')	204
Array and its	negative ('plus minus')	583
Number and its	negative ('plus minus')	621
	Negative infinity; the smallest representable value	331
Leftmost	neighboring elements	385
Rightmost	neighboring elements	384
Leftmost	neighboring elements cyclically	537
Rightmost	neighboring elements cyclically	536
None,	neither	355
Adding a	new dimension after dimension G Y-fold	10
Vector to expand a	new element after each one in X	507
Changing X if a	new input value is given	92
Duplicating element of X belonging to Y,1↑X until	next found	297
Extending a distance table to	next leg	224
Test if an element differs from the	next one	431
Transforming connection matrix X into a	node matrix	148
Transforming	node matrix X into a connection matrix	157
Index of first	non-blank, counted from the rear	79
Indices of last	non-blanks in rows	217
Corner element of a (	non-empty) array	615
Reshaping	non-empty lower-rank array X into a matrix	392
Reshaping	non-empty lower-rank array X into a matrix	402
Rows of	non-empty matrix X starting with a character in Y	499
	None, neither	355
Removing duplicate elements (	nub)	154
Removing duplicate elements (	nub)	501
Conversion of each row into a	number (default zero)	93
	Number and its negative ('plus minus')	621
	Number of columns in array X	599
	Number of columns in matrix X	410

Number of columns in matrix X 445
Number of columns in matrix X 600
Number of combinations of X objects taken Y at a time 142
Number of days in month X of year Y 74
Number of decimals (up to Y) of elements of X 149
Number of decimals of elements of X 87
Number of digit positions in integers in X 455
Number of digit positions in scalar X (depends on ⎕PP) 111
Number of digitpositions in integers in X 452
Number of digits in positive integers in X 456
Number of elements (also of a scalar) 366
Number of elements (also of a scalar) 596
Number of elements satisfying condition X 370
Number of occurrences of scalar X in array Y 236
Number of occurrences of scalar X in array Y 362
Number of permutations of X objects taken Y at a time 135
Number of rows in array X (also of a vector) 328
Number of rows in matrix X 411
Number of rows in matrix X 446
Number of rows in matrix X 601
Number of rows in matrix X 613
Binary format of decimal number X 45
Poisson distribution of states X with average number Y 140
Locating field number Y starting with first element of X 283
X first figurate numbers 260
Interpretation of roman numbers 342
X first triangular numbers 302
Integer and fractional parts of positive numbers 62
Randomizing random numbers (in ⎕LX in a workspace) 245
Ascending cardinal numbers (ranking, all different) 13
Descending cardinal numbers (ranking, all different) 17
Ascending cardinal numbers (ranking, shareable) 1
Arithmetic progression of Y numbers from X with step G 557
Sum of numbers in character matrix X 101
Ordinal numbers of words in X that indices Y point to 177
Y-shaped array of random numbers within (X[1],X[2]] 121
Leading zeroes for positive numbers X in fields of width Y 106
Test if numeric 400
Test if numeric 486
Transformation of alphanumeric string into numeric 78
Test if numeric and converting into numeric form 95
First part of numeric code ABBB 459
Last part of numeric code ABBB 479
Decoding numeric code ABBCCC into a matrix 63
Changing numeric code X into corresponding name in Y 554
Converting characters into numeric codes 561
Giving a numeric default value for input 97
Test if numeric and converting into numeric form 95
Converting expression X to numeric form with default value Y 94
Numeric headers (elements of X) for rows of table Y 102
Output of assigned numeric value 627
Empty numeric vector 563
Reshaping numeric vector X into a one-column matrix 203
Formatting a numerical vector to run down the page 103
Choosing Y objects out of ⍳X with replacement (roll) 122
Choosing Y objects out of ⍳X without replacement (deal) 123
Number of permutations of X objects taken Y at a time 135
Number of combinations of X objects taken Y at a time 142
Index of last occurrence 531
Index of last occurrence, counted from the rear 532
Index of first occurrence of elements of Y 338
Index of first occurrence of G in X (circularly) after Y 471
First occurrence of string X in matrix Y 229
First occurrence of string X in string Y 178
Indices of all occurrences of elements of Y in X 495
Number of occurrences of scalar X in array Y 236

Number of	occurrences of scalar X in array Y 362
	Occurrences of string X in string Y 147
	Occurrences of string X in string Y 176
	Occurrences of string X in string Y 219
Indices of all	occurrences of Y in X 503
Test if each element of X	occurs only once 353
Turn	off all ones after first one 307
Turn	off all ones after first zero 306
Test if each element of X occurs only	once 353
Turn off all ones after first	one 307
Turn on all zeroes after first	one 304
Test if an element differs from the next	one 431
Test if an element differs from the previous	one 432
Conditional change of elements of Y to	one according to X 569
Index of first	one after index Y in X 438
Reshaping vector X into a	one-column matrix 534
Reshaping vector X into a	one-column matrix 594
Reshaping numeric vector X into a	one-column matrix 203
Reshaping	one-element vector X into a scalar 371
Vector to expand a new element after each	one in X 507
Dividing a 400×12 character matrix into	one page 524
List of names in X (	one per row) 441
Reshaping vector X into a	one-row matrix 595
First group of	ones 251
First ones in groups of	ones 381
Last ones in groups of	ones 439
First ones in groups of	ones 440
Joining pairs of	ones 303
Places between pairs of	ones 305
Turn off all	ones after first one 307
Turn off all	ones after first zero 306
Last	ones in groups of ones 439
First	[redacted] in groups of ones 381
First	ones in groups of ones 440
Boolean vector of length Y with	ones in locations X 543
Indices of	ones in logical vector X 41
Indices of	ones in logical vector X 280
Indices of	ones in logical vector X 506
Lengths of groups of	ones in X 415
Groups of	ones in Y pointed by X 292
Groups of	ones in Y pointed by X (or trailing parts) 256
First	ones of subvectors of Y indicated by X (<\) 421
Matrix with X[I] leading	ones on row I 172
Matrix with X[I] trailing	[redacted] on row I 167
Vector of length Y with X	ones on the left, the rest zeroes 407
Test if each element of X occurs	only once 353
Centering character array X with	only right edge ragged 289
	Open a gap of X[I] after Y[G[I]] (for all I) 264
	Open a gap of X[I] before Y[G[I]] (for all I) 271
Exclusive	or 573
Continuing from line Y (if X>0)	or exit 120
	Or-reduction (∨/) of subvectors of Y indicated by X 491
Test if single	or scalar 546
	Or-scan (∨\) over subvectors of Y indicated by X 289
Zero	or space depending on the type of X (fill element) 409
Groups of ones in Y pointed by X (	or trailing parts) 256
Sorting X into ascending	order 35
Sorting X into descending	order 44
Sorting a matrix into lexicographic	order 19
Sorting rows of matrix X into ascending	order 8
Sorting rows of matrix X into ascending	order 9
Sorting a matrix into reverse lexicographic	order 38
Test if X is in ascending	order along direction Y 268
Test if X is in descending	order along direction Y 270
Playing	order in a cup for X ranked players 389
Hilbert matrix of	order X 198

Pascal's triangle of	order X (binomial coefficients)	136
Removing duplicate rows from	ordered matrix X	327
Removing duplicates from an	ordered vector	422
	Ordinal numbers of words in X that indices Y point to	177
Selection depending on index	origin	604
Selection depending on index	origin	609
Indexing independent of index	origin	606
Index	origin (⎕IO) as a vector	560
Changing an index	origin dependent argument to act as ⎕IO=1	626
Changing an index	origin dependent argument to act as ⎕IO=0	628
Changing an index	origin dependent result to be as ⎕IO=1	568
Changing an index	origin dependent result to be as ⎕IO=0	625
Test if X and Y are permutations of each	other	4
Selecting elements satisfying condition Y, zeroing	others	622
Choosing Y objects	out of ⍳X with replacement (roll)	122
Choosing Y objects	out of ⍳X without replacement (deal)	123
Conditional branch	out of program	118
Conditional branch	out of programs	89
Spacing	out text	490
Spacing	out text	580
Selecting specific elements from a 'large'	outer product	194
	Output of an empty line	617
	Output of assigned numeric value	627
	Output of assigned value	630
Sums	over (+/) subvectors of X, lengths in Y	257
Sums	over (+/) subvectors of Y indicated by X	291
Sum	over elements of X determined by elements of Y	150
Summation	over subsets of X specified by Y	241
Products	over subsets of X specified by Y	244
Cumulative sum (+\)	over subvectors of Y indicated by X	255
Maximum scan (⌈\)	over subvectors of Y indicated by X	2
Minimum scan (⌊\)	over subvectors of Y indicated by X	3
Or-scan (∨\)	over subvectors of Y indicated by X	289
And-scan (∧\)	over subvectors of Y indicated by X	290
Running parity (≠\)	over subvectors of Y indicated by X	254
Barchart of integer values (down the	page)	166
Barchart of integer values (across the	page)	171
Barchart of X with height Y (across the	page)	170
Barchart of two integer series (across the	page)	68
Histogram (distribution barchart; down the	page)	144
Dividing a 400×12 character matrix into one	page	524
Formatting a numerical vector to run down the	page	103
All	pairs of elements of ⍳X and ⍳Y	58
Joining	pairs of ones	303
Places between	pairs of ones	305
Index	pairs of saddle points	48
	Pairwise differences of successive columns (inverse of +\)	383
	Pairwise differences of successive elements of vector X	420
	Pairwise ratios of successive elements of vector X	419
Depth of	parentheses	295
Avoiding	parentheses with help of reversal	535
	Parity	361
Running	parity	309
Running	parity (≠\) over subvectors of Y indicated by X	254
Magnitude of fractional	part	465
Fractional	part	478
Last	part of numeric code ABBB	479
First	part of numeric code ABBB	459
Fractional	part with sign	476
Groups of ones in Y pointed by X (or trailing	parts)	256
Integer and fractional	parts of positive numbers	62
	Pascal's triangle of order X (binomial coefficients)	136
List of names in X (one	per row)	441
Row-by-row formatting (width G) of X with Y decimals	per row	109
Annuity coefficient: X	periods at interest rate Y %	186
Inverting a	permutation	37

Inverting a	*permutation* 550
Inverting a	*permutation* 553
Test if X is a	*permutation vector* 20
Test if X is a	*permutation vector* 158
Test if X is a	*permutation vector* 346
Test if X and Y are	*permutations of each other* 4
Number of	*permutations of X objects taken Y at a time* 135
	Places between pairs of ones 305
Transpose of	*planes of a rank three array* 527
Playing order in a cup for X ranked	*players* 389
	Playing order in a cup for X ranked players 389
Indexing with boolean value X (	*plotting a curve)* 605
Array and its negative ('	*plus minus')* 204
Array and its negative ('	*plus minus')* 583
Number and its negative ('	*plus minus')* 621
Value of saddle	*point* 262
Ordinal numbers of words in X that indices Y	*point to* 177
Value of polynomial with coefficients Y at	*point X* 83
Value of polynomial with coefficients Y at	*point X* 227
Value of Taylor series with coefficients Y at	*point X* 137
Value of Taylor series with coefficients Y at	*point X* 281
Groups of ones in Y	*pointed by X* 292
Groups of ones in Y	*pointed by X (or trailing parts)* 256
Ending points for X in indices	*pointed by Y* 529
Starting points for X in indices	*pointed by Y* 540
Index pairs of saddle	*points* 48
G-degree polynomial (curve) fit of	*points (X,Y)* 126
Cofficients of exponential (curve) fit of	*points (X,Y)* 127
Best linear fit of	*points (X,Y) (least squares)* 128
Ending	*points for X fields of width Y* 404
Starting	*points for X fields of width Y* 405
Ending	*points for X in indices pointed by Y* 529
Starting	*points for X in indices pointed by Y* 540
Ending	*points of equal groups* 417
Starting	*points of equal groups* 418
Contour levels Y at	*points with altitudes X* 179
Value of polynomial with coefficients Y at	*points X* 69
Graph of F(X) at	*points X ('X'∊F)* 90
Move set of	*points X into first quadrant* 174
	Poisson distribution of states X with average number Y 140
G-degree	*polynomial (curve) fit of points (X,Y)* 126
Value of	*polynomial with coefficients Y at points X* 69
Value of	*polynomial with coefficients Y at point X* 83
Value of	*polynomial with coefficients Y at point X* 227
	Polynomial with roots X 47
Derivate of	*polynomial X* 430
Product of two	*polynomials with coefficients X and Y* 163
'	*Position' matrix of a vector* 143
Helps locating column	*positions 1..X* 46
Number of digit	*positions in integers in X* 455
Number of digit	*positions in scalar X (depends on ⎕PP)* 111
Starting	*positions of subvectors (lengths in X)* 296
	Positive infinity; the largest representable value 337
Removing duplicates from	*positive integers* 505
Quick membership (∊) for	*positive integers* 333
Number of digits in	*positive integers in X* 456
Converting set of	*positive integers X into a mask* 329
	Positive maximum, at least zero (also for empty X) 334
Integer and fractional parts of	*positive numbers* 62
Leading zeroes for	*positive numbers X in fields of width Y* 106
Arithmetic	*precision of the system (in decimals)* 450
	Predicted values of best linear fit (least squares) 125
	Predicted values of exponential (curve) fit 124
Case structure with logical switch (	*preferring from start)* 115
	Present value of cash flows X at interest rate Y % 77
Test if an element differs from the	*previous one* 432

All primes up to X 175
Compound interest for principals Y at rates G % in times X 146
Entering from terminal data exceeding input (printing) width 598
Binomial distribution of X trials with probability Y 138
Alternating product 367
Shur product 188
Matrix product 240
Direct matrix product 187
Selecting specific elements from a 'large' outer product 194
Product of elements of X 368
Product of two polynomials with coefficients X and Y 163
Vector (cross) product of vectors 528
Scalar (dot) product of vectors 243
Products over subsets of X specified by Y 244
Conditional branch out of program 118
Conditional branch out of programs 89
Consecutive integers from X to Y (arithmetic progression) 558
Arithmetic progression from X to Y with step G 451
Arithmetic progression of Y numbers from X with step G 557
Removing punctuation characters 496
Conversion of characters to hexadecimal representation (⎕AV) 56
Decommenting a matrix representation of a function (⎕CR) 258
Changing an index origin dependent result to be as ⎕IO=1 568
Changing an index origin dependent result to be as ⎕IO=0 625
Changing an index origin dependent argument to act as ⎕IO=1 626
Changing an index origin dependent argument to act as ⎕IO=0 628
Index origin (⎕IO) as a vector 560
Randomizing random numbers (in ⎕LX in a workspace) 245
Number of digit positions in scalar X (depends on ⎕PP) 111
Move set of points X into first quadrant 174
(Real) solution of quadratic equation with coefficients X 363
Quick membership (∊) for positive integers 333
Locations of texts between quotes 288
Locations of texts between quotes 294
Locations of texts between and including quotes 287
Locations of texts between and including quotes 293
Doubling quotes (for execution) 25
Transformation from degrees to radians 134
Rotation matrix for angle X (in radians) counter-clockwise 132
Transformation from radians to degrees 133
Centering character array X with only right edge ragged 269
Centering character array X with ragged edges 248
Randomizing random numbers (in ⎕LX in a workspace) 245
Y-shaped array of random numbers within (X[1],X[2]] 121
Randomizing random numbers (in ⎕LX in a workspace) 245
Check for input in range 1..X 542
Test if X is an integer within range [G,H) 221
Test if X is within range [Y[1;],Y[2;]) 233
Test if X is within range (Y[1;],Y[2;]] 234
Test if X is within range (Y[1;],Y[2;]] 235
Test if X is within range [Y[1],Y[2]) 180
Test if X is within range (Y[1],Y[2]) 564
Test if X is within range [Y[1],Y[2]] 565
Size of range of elements of X 312
Definite integral of F(X) in range Y with G steps ('X'∊F) 91
Reshaping non-empty lower- rank array X into a matrix 392
Reshaping non-empty lower- rank array X into a matrix 402
Indexing when rank is not known beforehand 100
Rank of array X 612
Increasing rank of Y to rank of X 590
Increasing rank of Y to rank of X 590
Transpose of planes of a rank three array 527
Replicating a dimension of rank three array X Y-fold 200
Reduction in dimension Y, rank unchanged 502
Playing order in a cup for X ranked players 389
Ascending cardinal numbers (ranking, all different) 13

Descending cardinal numbers (ranking, all different) 17
Ascending cardinal numbers (ranking, shareable) 1
Future value of cash flows X at interest rate Y % 82
Present value of cash flows X at interest rate Y % 77
Annuity coefficient: X periods at interest rate Y % 186
Compound interest for principals Y at rates G % in times X 146
Pairwise ratios of successive elements of vector X 419
Temporary ravel of X for indexing with G 584
Temporary ravel of X for indexing with G 585
Conversion of indices to indices of raveled array 81
(Real) solution of quadratic equation with coefficients X 363
Index of first non-blank, counted from the rear 79
Index of last occurrence, counted from the rear 532
Sum of reciprocal series Y÷X 239
Sum of alternating reciprocal series Y÷X 238
And- reduction (∧/) of subvectors of Y indicated by X 492
Or- reduction (∨/) of subvectors of Y indicated by X 491
Reduction in dimension Y, rank unchanged 502
Reduction with a without respect to shape 511
Extending a transitive binary relation 230
Removing blank columns 226
Removing blank rows 225
Removing blanks 510
Removing columns Y from array X 396
Removing duplicate blank columns 220
Removing duplicate blank rows 216
Removing duplicate elements (nub) 154
Removing duplicate elements (nub) 501
Removing duplicate rows 70
Removing duplicate rows 206
Removing duplicate rows 208
Removing duplicate rows from ordered matrix X 327
Removing duplicate Y's from vector X 426
Removing duplicates from an ordered vector 422
Removing duplicates from positive integers 505
Removing elements Y from beginning of vector X 435
Removing elements Y from beginning and end of vector X 416
Removing elements Y from end of vector X 427
Removing elements Y from vector X 509
Removing every second element of X 469
Removing every Y:th element of X 466
Removing leading, multiple and trailing Y's 379
Removing leading and trailing blanks 259
Removing leading blank columns 210
Removing leading blank rows 211
Removing leading blanks 267
Removing leading blanks 299
Removing leading zeroes 437
Removing multiple blanks 424
Removing punctuation characters 496
Removing trailing blank columns 209
Removing trailing blank columns 272
Removing trailing blank rows 205
Removing trailing blanks 73
Removing trailing blanks 266
Removing trailing blanks 298
Removing trailing blanks 397
Removing trailing blanks 428
Repeat matrix 182
Executing costly monadic function F on repetitive arguments 151
Formatting with zero values replaced by blanks 108
Choosing Y objects out of ⍳X without replacement (deal) 123
Choosing Y objects out of ⍳X with replacement (roll) 122
Replacing all values Y[1] by Y[2] in X 425
Replacing elements of X in set Y by blanks / zeroes 480
Replacing elements of X not fulfilling Y by blanks / zeroes 483

Replacing elements of X not in set Y by blanks / zeroes 481
Replacing elements of Y satisfying X with G 504
Replacing first element of X with Y 434
Replacing last element of X with Y 433
Replicate Y[i] X[i] times (for all i) 250
Replicate Y[i] X[i] times (for all i) 251
Replicate Y[i] X[i] times (for all i) 253
Replicating a dimension of rank three array X Y-fold 200
Positive infinity: the largest representable value 337
Negative infinity: the smallest representable value 331
Conversion of characters to hexadecimal representation (⎕AV) 56
Decommenting a matrix representation of a function (⎕CR) 258
Representation of current date (descending format) 65
Representation of current date (ascending format) 104
Representation of current date (American) 107
Representation of current time (24 hour clock) 64
Representation of current time (12 hour clock) 105
Integer representation of logical vectors 84
Representation of X in base Y 54
All binary representations up to X (truth table) 52
All binary representations with X bits (truth table) 60
Reshaping non-empty lower-rank array X into a matrix 392
Reshaping non-empty lower-rank array X into a matrix 402
Reshaping numeric vector X into a one-column matrix 203
Reshaping one-element vector X into a scalar 371
Reshaping vector X into a one-column matrix 534
Reshaping vector X into a one-column matrix 594
Reshaping vector X into a one-row matrix 595
Reshaping vector X into a two-column matrix 591
Reshaping X into a scalar 616
Reduction with ⍺ without respect to shape 511
Vector of length Y with X ones on the left, the rest zeroes 407
Restoring argument of cumulative sum (inverse of +\) 442
Changing an index origin dependent result to be as ⎕IO=1 568
Changing an index origin dependent result to be as ⎕IO=0 625
Evaluation of several expressions: results form a vector 99
Retrieving scattered elements Y from matrix X 522
Avoiding parentheses with help of reversal 535
Reversal (⌽) of subvectors having lengths Y 39
Reversal (⌽) of subvectors starting at indices X 40
Sorting a matrix into reverse lexicographic order 38
Reverse vector X on condition Y 533
Justifying right 76
Justifying right 275
Transferring text X to right edge of field of width Y 412
Centering character array X with only right edge ragged 269
Justifying right fields of X (lengths Y) to length G 184
Shifting vector X right with Y without rotate 386
Rightmost neighboring elements 384
Rightmost neighboring elements cyclically 536
Choosing Y objects out of ⍳X with replacement (roll) 122
Interpretation of roman numbers 342
Polynomial with roots X 47
Shifting vector X left with Y without rotate 387
Shifting vector X right with Y without rotate 386
Rotate first elements (1⌽) of subvectors of Y indicated by X 2
Rotation matrix for angle X (in radians) counter-clockwise 132
Rounding, to nearest even integer for .5 = 1|X 454
Rounding to nearest even integer 453
Rounding to nearest hundredth 461
Rounding to nearest integer 462
Rounding to X decimals 460
Rounding to zero values of X close to zero 474
List of names in X (one per row) 441
Moving all blanks to end of each row 160
Row-by-row formatting (width G) of X with Y decimals per row 10

Evaluating a two- row determinant 313
Evaluating a two- row determinant 314
Row-by- row formatting (width G) of X with Y decimals per row 109
Inserting vector X into matrix Y after row G 375
Matrix with X[i] leading ones on row i 172
Matrix with X[i] trailing ones on row i 167
Matrix with X[i] leading zeroes on row i 168
Matrix with X[i] trailing zeroes on row i 165
Conversion of each row into a number (default zero) 93
Adding an empty row into X after row Y 487
Adding an empty row into X after rows Y 485
Forming a two- row matrix 588
Reshaping vector X into a one- row matrix 595
Forming a Y- row matrix with all rows alike (X) 593
Forming first row of a matrix to be expanded 408
Test if vector Y is a row of array X 228
Multiplication of each row of X by vector Y 611
Adding X to each row of Y 189
Adding X to each row of Y 190
Row sum of a matrix 373
Adding an empty row into X after row Y . 487
Input of row Y of text matrix X 378
ow-by-row formatting (width G) of X with Y decimals per row 109
Removing duplicate rows 70
Removing duplicate rows 206
Removing duplicate rows 208
Comparison of successive rows 344
Blank rows 359
Removing duplicate blank rows 218
Removing trailing blank rows 205
Removing leading blank rows 211
Removing blank rows 225
Indices of last non-blanks in rows 217
Array with shape of Y and X as its rows 614
Applying to columns function F defined on rows 514
Forming a Y-row matrix with all rows alike (X) 593
'X⍳Y' by rows for elements of matrices 153
Drop of Y first rows from matrix X 388
Drop of Y first rows from matrix X 444
Removing duplicate rows from ordered matrix X 327
Number of rows in array X (also of a vector) 328
Number of rows in matrix X 411
Number of rows in matrix X 446
Number of rows in matrix X 601
Number of rows in matrix X 613
Comparing vector Y with rows of array X 232
Indices of first blanks in rows of array X 282
'X⍳Y' for rows of matrices 207
Average (mean value) of rows of matrix X 325
Sorting rows of matrix X into ascending order 8
Sorting rows of matrix X into ascending order 9
Rows of matrix X starting with string Y 216
Rows of non-empty matrix X starting with a character in Y 499
Numeric headers (elements of X) for rows of table Y 102
Sorting rows of X according to key Y (alphabetizing) 18
Test if rows of X contain elements differing from Y 231
Adding an empty row into X after rows Y 485
Formatting a numerical vector to run down the page 103
Running parity 309
Running parity (≠\) over subvectors of Y indicated by X 254
Running sum of Y consecutive elements of X 285
Value of saddle point 262
Index pairs of saddle points 48
Alphabetizing X; equal alphabets in same column of Y 395
Index of first satisfied condition in X 559
Number of elements satisfying condition X 370

Selecting elements of X	satisfying condition Y	512
Selecting elements	satisfying condition Y, zeroing others	622
Moving elements	satisfying condition Y to the start of X	43
Replacing elements of Y	satisfying X with G	504
Reshaping X into a	scalar	616
Test if single or	scalar	546
Number of elements (also of a	scalar)	366
Number of elements (also of a	scalar)	596
Reshaping one-element vector X into a	scalar	371
	Scalar (dot) product of vectors	243
Number of digit positions in	scalar X (depends on ⎕PP)	111
Number of occurrences of	scalar X in array Y	236
Number of occurrences of	scalar X in array Y	362
Adding	scalar Y to last element of X	406
And-	scan (∧\) over subvectors of Y indicated by X	290
Minimum	scan (⌊\) over subvectors of Y indicated by X	3
Maximum	scan (⌈\) over subvectors of Y indicated by X	2
Or-	scan (∨\) over subvectors of Y indicated by X	289
	Scan from end with ⍺	484
Indexing	scattered elements	80
Retrieving	scattered elements Y from matrix X	522
Removing every	second element of X	469
	Selecting elements of X satisfying condition Y	512
	Selecting elements satisfying condition Y, zeroing others	622
	Selecting specific elements from a 'large' outer product	194
	Selection between X and Y depending on condition G	567
	Selection depending on index origin	504
	Selection depending on index origin	609
Joining	sentences	597
Digits of X	separately	53
	Separating a date YYMMDD to YY, MM, DD	57
	Separating variable length lines	579
Alternating	series (1 ¯1 2 ¯2 3 ¯3 ...)	301
Barchart of two integer	series (across the page)	68
Interpolated value of	series (X,Y) at G	67
Value of Taylor	series with coefficients Y at point X	137
Value of Taylor	series with coefficients Y at point X	281
Sum of reciprocal	series Y+X	239
Sum of alternating reciprocal	series Y+X	238
Move	set of points X into first quadrant	174
Converting	set of positive integers X into a mask	329
Which elements of X are not in	set Y (difference of sets)	552
Replacing elements of X in	set Y by blanks / zeroes	480
Replacing elements of X not in	set Y by blanks / zeroes	481
Union of	sets, ∪	497
Intersection of	sets, ∩	500
Identity of two	sets	345
Elements of X not in Y (difference of	sets)	498
Which elements of X are not in set Y (difference of	sets)	552
	Setting latent expression	631
Evaluation of	several expressions; results form a vector	99
Reduction with ⍺ without respect to	shape	511
Zero array of	shape and size of X	624
Test if arrays of equal	shape are identical	357
Matrix with	shape of Y and X as its columns	517
Array with	shape of Y and X as its rows	614
Y-	shaped array of random numbers within (X[1],X[2]]	121
Ascending cardinal numbers (ranking,	shareable)	1
	Shifting vector X left with Y without rotate	387
	Shifting vector X right with Y without rotate	386
	Shur product	188
Area of triangle with	side lengths in X (Heron's formula)	318
Fractional part with	sign	476
Conditional elementwise change of	sign	603
Conditional elementwise change of	sign	623
Square of elements of X without change of	sign	477

Increasing absolute value without change of	sign	475
Conditional branch depending on	sign of X	119
Choosing according to	signum	602
Complementary angle (arccos	sin X)	131
A way to combine trigonometric functions (	sin X cos Y)	130
Test if	single or scalar	546
Zeroing a vector (without change of	size)	608
lassification of elements Y into X classes of equal	size	156
	Size of range of elements of X	312
Zero array of shape and	size of X	624
Negative infinity; the	smallest representable value	331
(Real)	solution of quadratic equation with coefficients X	363
	Sorting a matrix according to Y:th column	33
	Sorting a matrix into lexicographic order	19
	Sorting a matrix into reverse lexicographic order	38
Choosing	sorting direction during execution	34
	Sorting indices X according to data Y	32
	Sorting rows of matrix X into ascending order	8
	Sorting rows of matrix X into ascending order	9
	Sorting rows of X according to key Y (alphabetizing)	18
	Sorting subvectors of lengths Y	5
	Sorting X into ascending order	35
	Sorting X into descending order	44
	Sorting Y according to X	36
Zero or	space depending on the type of X (fill element)	409
	Spacing out text	490
	Spacing out text	580
Selecting	specific elements from a 'large' outer product	194
Summation over subsets of X	specified by Y	241
Products over subsets of X	specified by Y	244
Maxima of elements of subsets of X	specified by Y	213
Test for symmetricity of	square matrix X	520
Test for antisymmetricity of	square matrix X	519
	Square of elements of X without change of sign	477
Best linear fit of points (X,Y) (least	squares)	128
Predicted values of best linear fit (least	squares)	125
Sum of	squares of X	242
Sum of	squares of X	246
	Standard deviation of X	319
Case structure with logical switch (preferring from	start)	115
Moving elements satisfying condition Y to the	start of X	43
Reversal (⌽) of subvectors	starting at indices X	40
Indices of subvectors of length Y	starting at X+1	202
Changing lengths of subvectors to	starting indicators	278
Changing	starting indicators of subvectors to lengths	423
	Starting points for X fields of width Y	405
	Starting points for X in indices pointed by Y	540
	Starting points of equal groups	418
	Starting positions of subvectors (lengths in X)	296
Rows of non-empty matrix X	starting with a character in Y	499
Locating field number Y	starting with first element of X	283
Rows of matrix X	starting with string Y	215
Poisson distribution of	states X with average number Y	140
Arithmetic progression from X to Y with	step G	451
Arithmetic progression of Y numbers from X with	step G	557
Definite integral of F(X) in range Y with G	steps ('X'∊F)	91
Fifo	stock Y decremented with X units	286
Syntax error to	stop execution	629
Underlining a	string	566
Choosing a	string according to boolean value G	493
Transformation of alphanumeric	string into numeric	78
First occurrence of	string X in matrix Y	229
First occurrence of	string X in string Y	178
Occurrences of	string X in string Y	147
Occurrences of	string X in string Y	176
Occurrences of	string X in string Y	219

Occurrences of string X in string Y 147
Occurrences of string X in string Y 176
Occurrences of string X in string Y 219
Rows of matrix X starting with string Y 216
First occurrence of string X in string Y 178
Case structure according to key vector G 117
Case structure with an encoded branch destination 66
Case structure with integer switch 116
Case structure with logical switch (preferring from start) 115
Test if Y is a subset of X (Y ⊂ X) 351
Matrix for choosing all subsets of X (truth table) 59
Maxima of elements of subsets of X specified by Y 213
Summation over subsets of X specified by Y 241
Products over subsets of X specified by Y 244
Insert vector for X[i] zeroes after i:th subvector 265
G:th subvector of Y (subvectors indicated by X) 300
Starting positions of subvectors (lengths in X) 296
Reversal (⌽) of subvectors having lengths Y 39
G:th subvector of Y (subvectors indicated by X) 300
Indices of subvectors of length Y starting at X+1 202
Sorting subvectors of lengths Y 5
Sums over (+/) subvectors of X, lengths in Y 257
Sums over (+/) subvectors of Y indicated by X 291
Or-reduction (∨/) of subvectors of Y indicated by X 491
And-reduction (∧/) of subvectors of Y indicated by X 492
First ones of subvectors of Y indicated by X (<\) 421
Minima (⌊/) of elements of subvectors of Y indicated by X 6
Maxima (⌈/) of elements of subvectors of Y indicated by X 14
Grading down subvectors of Y indicated by X 15
Grading up subvectors of Y indicated by X 7
Maximum scan (⌈\) over subvectors of Y indicated by X 2
Minimum scan (⌊\) over subvectors of Y indicated by X 3
Running parity (≠\) over subvectors of Y indicated by X 254
Cumulative sum (+\) over subvectors of Y indicated by X 255
Or-scan (∨\) over subvectors of Y indicated by X 289
And-scan (∧\) over subvectors of Y indicated by X 290
Rotate first elements (1⌽) of subvectors of Y indicated by X 21
Reversal (⌽) of subvectors starting at indices X 40
Changing lengths of subvectors to ending indicators 277
Changing starting indicators of subvectors to lengths 423
Changing lengths of subvectors to starting indicators 278
Lengths of subvectors with equal elements 413
Sum elements of X marked by succeeding identicals in Y 284
(Cyclic) compression of successive blanks 494
Pairwise differences of successive columns (inverse of +\) 383
Pairwise differences of successive elements of vector X 420
Pairwise ratios of successive elements of vector X 419
Differences of successive elements of X along direction Y 436
Comparison of successive rows 344
Successive transposes of G (X after Y: X⍉Y⍉G) 523
Cumulative sum 310
Alternating sum 369
Matrix sum 191
Cumulative sum (+\) over subvectors of Y indicated by X 255
Restoring argument of cumulative sum (inverse of +\) 442
Sum elements of X marked by succeeding identicals in Y 284
Column sum of a matrix 372
Row sum of a matrix 373
Sum of alternating reciprocal series Y+X 238
Sum of elements of X 374
Sum of numbers in character matrix X 101
Sum of reciprocal series Y+X 239
Sum of squares of X 242
Sum of squares of X 246
Running sum of Y consecutive elements of X 285
Sum over elements of X determined by elements of Y 150

Summation over subsets of X specified by Y   241
Sums according to codes G   152
Sums over (+/) subvectors of X, lengths in Y   257
Sums over (+/) subvectors of Y indicated by X   291
Case structure with integer switch   116
Case structure with logical switch (preferring from start)   115
Test for symmetricity of matrix X   86
Test for symmetricity of square matrix X   520
Syntax error to stop execution   629
Arithmetic precision of the system (in decimals)   450
Maximum table   183
Multiplication table   199
Comparison table   169
Divisibility table   164
All binary representations up to X (truth table)   52
Matrix for choosing all subsets of X (truth table)   59
All binary representations with X bits (truth table)   60
Extending a distance table to next leg   224
Numeric headers (elements of X) for rows of table Y   102
Take of at most X elements from Y   394
Number of permutations of X objects taken Y at a time   135
Number of combinations of X objects taken Y at a time   142
Taking every Y:th element of X   467
Value of Taylor series with coefficients Y at point X   137
Value of Taylor series with coefficients Y at point X   281
Handling array X temporarily as a vector   592
Temporary ravel of X for indexing with G   584
Temporary ravel of X for indexing with G   585
Entering from terminal data exceeding input (printing) width   598
Test for antisymmetricity of matrix X   85
Test for antisymmetricity of square matrix X   519
Test for symmetricity of matrix X   86
Test for symmetricity of square matrix X   520
Test if all elements of vector X are equal   159
Test if all elements of vector X are equal   214
Test if all elements of vector X are equal   215
Test if all elements of vector X are equal   311
Test if all elements of vector X are equal   315
Test if all elements of vector X are equal   316
Test if all elements of vector X are equal   317
Test if all elements of vector X are equal   324
Test if all elements of vector X are equal   340
Test if all elements of vector X are equal   341
Test if all elements of vector X are equal   343
Test if all elements of vector X are equal   347
Test if all elements of vector X are equal   349
Test if all elements of vector X are equal   352
Test if all elements of vector X are equal   354
Test if all elements of vector X are equal   358
Test if an element differs from the next one   431
Test if an element differs from the previous one   432
Test if arrays are identical   544
Test if arrays of equal shape are identical   357
Test if each element of X occurs only once   353
Test if even   473
Test if integer   457
Test if numeric   400
Test if numeric   486
Test if numeric and converting into numeric form   95
Test if rows of X contain elements differing from Y   231
Test if single or scalar   546
Test if vector   547
Test if vector Y is a row of array X   228
Test if X and Y are permutations of each other   4
Test if X and Y have elements in common   348
Test if X is a leap year   463

	Test if X is a lower triangular matrix	162
	Test if X is a permutation vector	20
	Test if X is a permutation vector	158
	Test if X is a permutation vector	346
	Test if X is a valid APL name	322
	Test if X is an empty array	548
	Test if X is an integer within range [G,H)	221
	Test if X is an upper triangular matrix	161
	Test if X is boolean	350
	Test if X is in ascending order along direction Y	268
	Test if X is in descending order along direction Y	270
	Test if X is within range [Y[1],Y[2])	180
	Test if X is within range [Y[1;],Y[2;])	233
	Test if X is within range (Y[1;],Y[2;]]	234
	Test if X is within range (Y[1;],Y[2;]]	235
	Test if X is within range (Y[1],Y[2])	564
	Test if X is within range [Y[1],Y[2]]	565
	Test if Y is a subset of X (Y ∊ X)	351
Moving all blanks to end of	text	42
Conditional in	text	508
Spacing out	text	490
Spacing out	text	580
Centering	text line X into a field of width Y	323
Centering	text line X into a field of width Y	393
Input of row Y of	text matrix X	378
Transferring	text X to right edge of field of width Y	412
Locations of	texts between and including quotes	287
Locations of	texts between and including quotes	293
Locations of	texts between quotes	288
Locations of	texts between quotes	294
Ordinal numbers of words in X	that indices Y point to	177
Zeroing elements of Y depending on	their values	545
Transpose of planes of a rank	three array	527
Replicating a dimension of rank	three array X Y-fold	200
Number of combinations of X objects taken Y at a	time	142
Number of permutations of X objects taken Y at a	time	135
Representation of current	time (24 hour clock)	64
Representation of current	time (12 hour clock)	105
Duplicating vector X Y	times	610
Replicate Y[i] X[i]	times (for all i)	250
Replicate Y[i] X[i]	times (for all i)	251
Replicate Y[i] X[i]	times (for all i)	253
Compound interest for principals Y at rates G % in	times X	146
Removing	trailing blank columns	209
Removing	trailing blank columns	272
Removing	trailing blank rows	205
Removing leading and	trailing blanks	259
Removing	trailing blanks	73
Removing	trailing blanks	266
Removing	trailing blanks	298
Removing	trailing blanks	397
Removing	trailing blanks	428
Matrix with X[i]	trailing ones on row i	167
Groups of ones in Y pointed by X (or	trailing parts)	256
Removing leading, multiple and	trailing Y's	379
Matrix with X[i]	trailing zeroes on row i	165
	Transferring text X to right edge of field of width Y	412
	Transformation from degrees to radians	134
	Transformation from radians to degrees	133
	Transformation of alphanumeric string into numeric	78
	Transforming connection matrix X into a node matrix	148
	Transforming connectivity list X into a connectivity matrix	71
	Transforming connectivity matrix X into a connectivity list	50
	Transforming node matrix X into a connection matrix	157
Extending a	transitive binary relation	230
Forming a	transitive closure	112

Transpose matrix X on condition Y 518
Transpose of planes of a rank three array 527
Successive transposes of G (X after Y; X⍉(⍉G) 523
** Transpose* of matrix X with column fields of width Y* 515
Binomial distribution of X trials with probability Y 138
Pascal's triangle of order X (binomial coefficients) 136
Area of triangle with side lengths in X (Heron's formula) 318
Test if X is an upper triangular matrix 161
X×X upper triangular matrix 195
Test if X is a lower triangular matrix 162
X×X lower triangular matrix 196
X first triangular numbers 302
A way to combine trigonometric functions (sin X cos Y) 130
All binary representations up to X (truth table) 52
Matrix for choosing all subsets of X (truth table) 59
All binary representations with X bits (truth table) 60
Turn off all ones after first one 307
Turn off all ones after first zero 306
Turn on all zeroes after first one 304
Turn on all zeroes after first zero 308
Forming a two-column matrix 589
Reshaping vector X into a two-column matrix 591
Barchart of two integer series (across the page) 68
Product of two polynomials with coefficients X and Y 163
Evaluating a two-row determinant 313
Evaluating a two-row determinant 314
Forming a two-row matrix 588
Identity of two sets 345
Zero or space depending on the type of X (fill element) 409
Reduction in dimension Y, rank unchanged 502
Merging X, Y, Z ... under control of G (mesh) 11
Merging X and Y under control of G (mesh) 12
Merging X and Y under control of G (mesh) 16
Merging X and Y under control of G (mesh) 31
Merging X and Y under control of G (mesh) 482
Underlining a string 586
Underlining words 488
Changing index of an unfound element to zero 472
Changing index of an unfound element to zero (not effective) 541
Union of sets, ∪ 497
Fifo stock Y decremented with X units 286
Duplicating element of X belonging to Y,1↑X until next found 297
Grade up according to key Y 30
Grading up subvectors of Y indicated by X 7
All primes up to X 175
All binary representations up to X (truth table) 52
Number of decimals (up to Y) of elements of X 149
Incrementing cyclic counter X with upper limit Y 61
Test if X is an upper triangular matrix 161
X×X upper triangular matrix 195
Using a variable named according to X 88
Using boolean matrix Y in expanding X 489
Test if X is a valid APL name 322
Output of assigned numeric value 627
Output of assigned value 630
Arithmetic average (mean value) 365
Positive infinity; the largest representable value 337
Negative infinity; the smallest representable value 331
Arithmetic average (mean value), also for an empty array 332
Average (mean value) of columns of matrix X 326
Average (mean value) of elements of X along direction Y 364
Average (mean value) of rows of matrix X 325
Arithmetic average (mean value) of X weighted by Y 237
Giving a numeric default value for input 97
Giving a character default value for input 403
Limiting indices and giving a default value G 458

Choosing a string according to boolean	*value G* 493
Changing X if a new input	*value is given* 92
Future	*value of cash flows X at interest rate Y %* 82
Present	*value of cash flows X at interest rate Y %* 77
	Value of polynomial with coefficients Y at points X 69
	Value of polynomial with coefficients Y at point X 83
	Value of polynomial with coefficients Y at point X 227
	Value of saddle point 262
Interpolated	*value of series (X,Y) at G* 67
	Value of Taylor series with coefficients Y at point X 137
	Value of Taylor series with coefficients Y at point X 281
Increasing absolute	*value without change of sign* 475
Indexing with boolean	*value X (plotting a curve)* 605
Changing elements in X with	*value Y[1] into Y[2]* 380
Converting expression X to numeric form with default	*value Y* 94
Avoiding division by zero error (gets	*value zero)* 572
Zeroing all boolean	*values* 566
Zeroing elements of Y depending on their	*values* 545
Barchart of integer	*values (across the page)* 171
Barchart of integer	*values (down the page)* 166
Predicted	*values of best linear fit (least squares)* 125
Predicted	*values of exponential (curve) fit* 124
Assign	*values of expressions in X to variables named in Y* 98
Rounding to zero	*values of X close to zero* 474
Formatting with zero	*values replaced by blanks* 108
Replacing all	*values Y[1] by Y[2] in X* 425
Separating	*variable length lines* 579
Using a	*variable named according to X* 88
Assign values of expressions in X to	*variables named in Y* 98
	Variance (dispersion) of X 320
'Position' matrix of a	*vector* 143
Index origin (⎕IO) as a	*vector* 560
First column as a	*vector* 607
Empty numeric	*vector* 563
Test if	*vector* 547
Test if X is a permutation	*vector* 20
Test if X is a permutation	*vector* 158
Test if X is a permutation	*vector* 346
Handling array X temporarily as a	*vector* 592
Removing duplicates from an ordered	*vector* 422
Number of rows in array X (also of a	*vector)* 328
Evaluation of several expressions; results form a	*vector* 99
	Vector (cross) product of vectors 528
Zeroing a	*vector (without change of size)* 608
	Vector (X[1]ρ1),(X[2]ρ0),(X[3]ρ1),... 252
	Vector (X[1]ρ1),(Y[1]ρ0),(X[2]ρ1),... 247
	Vector (Y[1]+⍳X[1]),(Y[2]+⍳X[2]),(Y[3]+⍳X[3]),... 249
Insert	*vector for X[i] elements before i:th element* 279
Insert	*vector for X[i] zeroes after i:th subvector* 265
Case structure according to key	*vector G* 117
Boolean	*vector of length Y with ones in locations X* 543
	Vector of length Y with X ones on the left, the rest zeroes 407
Boolean	*vector of length Y with zeroes in locations X* 539
	Vector to expand a new element after each one in X 507
Formatting a numerical	*vector to run down the page* 103
Expansion	*vector with zero after indices Y* 27
Expansion	*vector with zero after indices Y* 538
All indices of	*vector X* 556
Indices of ones in logical	*vector X* 41
Indices of ones in logical	*vector X* 280
Indices of ones in logical	*vector X* 506
Removing duplicate Y's from	*vector X* 426
Removing elements Y from	*vector X* 509
Moving index of width Y for	*vector X* 201
Index of key Y in key	*vector X* 552
Removing elements Y from end of	*vector X* 427

Removing elements Y from beginning of vector X 435
Pairwise ratios of successive elements of vector X 419
Catenating G elements H after indices Y in vector X 28
Catenating G elements H before indices Y in vector X 29
Removing elements Y from beginning and end of vector X 416
Pairwise differences of successive elements of vector X 420
Field lengths of vector X; G ↔ ending indices 414
Inserting Y *'s into vector X after indices G 26
Test if all elements of vector X are equal 159
Test if all elements of vector X are equal 214
Test if all elements of vector X are equal 215
Test if all elements of vector X are equal 311
Test if all elements of vector X are equal 315
Test if all elements of vector X are equal 316
Test if all elements of vector X are equal 317
Test if all elements of vector X are equal 324
Test if all elements of vector X are equal 340
Test if all elements of vector X are equal 341
Test if all elements of vector X are equal 343
Test if all elements of vector X are equal 347
Test if all elements of vector X are equal 349
Test if all elements of vector X are equal 352
Test if all elements of vector X are equal 354
Test if all elements of vector X are equal 358
Reshaping numeric vector X into a one-column matrix 203
Reshaping vector X into a one-column matrix 534
Reshaping vector X into a one-column matrix 594
Reshaping vector X into a one-row matrix 595
Reshaping one-element vector X into a scalar 371
Reshaping vector X into a two-column matrix 591
Inserting vector X into matrix Y after row G 375
Shifting vector X left with Y without rotate 387
Reverse vector X on condition Y 533
Shifting vector X right with Y without rotate 386
Duplicating vector X Y times 610
Multiplication of each row of X by vector Y 611
Multiplication of each column of matrix X by vector Y 516
Test if vector Y is a row of array X 228
Inserting vector Y to the end of matrix X 376
Comparing vector Y with rows of array X 232
Scalar (dot) product of vectors 243
Vector (cross) product of vectors 528
Integer representation of logical vectors 84
Vectors as matrices in catenation 273
Vectors as matrices in catenation 274
Index of first differing element in vectors X and Y 551
Merging vectors X and Y alternately 578
A way to combine trigonometric functions (sin X cos Y) 130
Arithmetic average (mean value) of X weighted by Y 237
Maximum of X with weights Y 222
Minimum of X with weights Y 223
Indexing when rank is not known beforehand 100
Which class do elements of X belong to 181
Which elements of X are not in set Y (difference of sets) 552
Widening matrix X to be compatible with Y 391
Formatting X with H decimals in fields of width G 110
Row-by-row formatting (width G) of X with Y decimals per row 109
Division to Y classes with width H, minimum G 173
Ending points for X fields of width Y 404
Starting points for X fields of width Y 405
Centering text line X into a field of width Y 323
Centering text line X into a field of width Y 393
Transferring text X to right edge of field of width Y 412
"Transpose" of matrix X with column fields of width Y 515
Leading zeroes for positive numbers X in fields of width Y 106
Moving index of width Y for vector X 201

Entering from terminal data exceeding input (printing)	*width* 598
Editing X with Y ∇-	*wise* 263
Y-shaped array of random numbers	*within (X[1],X[2]]* 121
Test if X is an integer	*within range [G,H)* 221
Test if X is	*within range [Y[1;],Y[2;])* 233
Test if X is	*within range (Y[1;],Y[2;]]* 234
Test if X is	*within range (Y[1;],Y[2;]]* 235
Test if X is	*within range [Y[1],Y[2])* 180
Test if X is	*within range (Y[1],Y[2])* 564
Test if X is	*within range [Y[1],Y[2]]* 565
Increasing absolute value	*without change of sign* 475
Square of elements of X	*without change of sign* 477
Zeroing a vector (	*without change of size)* 608
Choosing Y objects out of ⍳X	*without replacement (deal)* 123
Reduction with ⍺	*without respect to shape* 511
Shifting vector X left with Y	*without rotate* 387
Shifting vector X right with Y	*without rotate* 386
First	*word in X* 401
Underlining	*words* 488
Ordinal numbers of	*words in X that indices Y point to* 177
Randomizing random numbers (in ⎕LX in a	*workspace)* 245
Test if X is a leap	*year* 463
Number of days in month X of	*year Y* 74
Separating a date YYMMDD to	*YY, MM, DD* 57
Separating a date	*YYMMDD to YY, MM, DD* 57
Turn off all ones after first	*zero* 306
Turn on all zeroes after first	*zero* 308
Rounding to zero values of X close to	*zero* 474
Changing index of an unfound element to	*zero* 472
Avoiding division by zero error (gets value	*zero)* 572
Conversion of each row into a number (default	*zero)* 93
Positive maximum, at least	*zero (also for empty X)* 334
Changing index of an unfound element to	*zero (not effective)* 541
Expansion vector with	*zero after indices Y* 27
Expansion vector with	*zero after indices Y* 538
	Zero array of shape and size of X 624
Avoiding division by	*zero error (gets value zero)* 572
	Zero or space depending on the type of X (fill element) 409
Rounding to	*zero values of X close to zero* 474
Formatting with	*zero values replaced by blanks* 108
Removing leading	*zeroes* 437
Replacing elements of X in set Y by blanks /	*zeroes* 480
Replacing elements of X not in set Y by blanks /	*zeroes* 481
Turn on all	*zeroes after first one* 304
Turn on all	*zeroes after first zero* 308
Insert vector for X[i]	*zeroes after i:th subvector* 265
Leading	*zeroes for positive numbers X in fields of width Y* 106
Boolean vector of length Y with	*zeroes in locations X* 539
Matrix with X[i] leading	*zeroes on row i* 168
Matrix with X[i] trailing	*zeroes on row i* 165
Replacing elements of X not fulfilling Y by blanks /	*zeroes* 483
Converting	*zeroes to Y* 574
Vector of length Y with X ones on the left, the rest	*zeroes* 407
	Zeroing a vector (without change of size) 608
	Zeroing all boolean values 566
	Zeroing elements of Y depending on their values 545
Selecting elements satisfying condition Y,	*zeroing others* 622